Evolution of Non-Functional Requirement a literature survey

Raghu Veer [1],
[1]Address CS&E Department BBDNITM,
Lucknow, India

Dr. Deepak Kumar Singh [2]
[2]Address Jaipuria Institute of Management,
Lucknow

***Abstract* - In current era there are several technologies used to achieve software qualities like requirement elicitation-based Metrix founder and optimizer adapter and many more. To perform any software related task the main aspect in developer side is to validate requirements like non-functional as well as functional requirements. Developers mainly focus on functional requirements and ignores the Non-Functional requirements but Non-Functional requirements are also need equivalent focus such as Security, Performance, Scalability, Capacity, Availability, Reliability, Recoverability, Maintainability, Serviceability etc. In the given paper author read more than 100 papers and find various issues which is beneficial in requirement gathering and comparative study of Non-Functional Requirements. In this survey paper the authors provide the importance and uses of the Non-Functional Requirements in different years.**

Keywords - Functional Requirements, Non Functional Requirements, Software Engineering, Software Requirements

INTRODUCTION

In the present scenario there is lot of uses requirement engineering. When we develop any software product then first aspect is find the actual requirement i.e. what is user want and what we want to delivered. In this survey paper author tries to read 100 research paper and provide a comprehensive survey in terms of non-functional requirement. Requirement: As we know that requirement engineering is most powerful weapon in software engineering. It is an important phase software development life cycle. In simple terms requirement definition is: what is user want or what is needs? There are many phases in requirement engineering like:

i. Requirement Elicitation

ii. Requirement Verification

iii. Requirement Validation

iv. Requirement refinements

v. Requirement monitoring

vi. Requirement Tracing

In requirement engineering there are two types of requirements:

1. Functional Requirements
2. Non-Functional Requirements

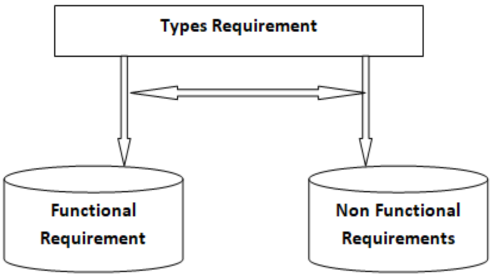


Fig.1. Types of requirement

1.1 Functional Requirements: In requirement Engineering Functional requirements related to functional aspects. Conceptually it is related to functionality of software system.

1.2 Non-Functional Requirements: Non-functional requirement is a part of requirement engineering. It is based on parameters which is related to quality Software quality is most important aspects to defining absolute values of software. So Non-functional requirements is focusing of this aspects, like security, fault tolerance, usability Nonfunctional requirement is a part of requirement engineering. It is based on parameters which is related to quality Software quality is most important aspects to defining absolute values of software. So Nonfunctional requirements is focusing of this aspect, like security, fault tolerance, usability

This paper analyses approx. 100 papers related to requirement engineering and extract the conclusion in all papers and provide that conclusion in authors survey paper. The conclusion is based on requirement and its description.

Table 1 Characteristics of NFR'S

Requirement	Description
QoS requirements	It is integrated with other services which is contain run time functionality
Development requirements	It contains service engineering process list and its information.
Compliance requirements	Compliance requirements relate to standards, rules and regulations.
Response time	Time between user send the request for service & a complete response is provided
Fault tolerance	How it can overcome errors, either hardware or software-based is One of the properties of the system. In the case of the cloud, non-software errors can be generated either on physical or virtual machines hosting the service.

Maintainability	(analyzability, changeability, stability and testability
Interoperability	SOA and its competent technologies promise cost savings due to their uninterrupted and tight integration and interactivity as well as reuse capabilities.
Portability	(Adaptability, instability, and coexistence): In SOA, portability has become less of a problem than traditional software applications.
Reusability	The degree to which the service can be used in more than one service application, without having to overarching, without much overhead to search, configure and invite it.
Integration	To integrate various services of a system..
Flexibility	Ability to adapt changing different requirements more efficiently, easily and quickly
Agility	for unexpected and unexpected changes the ability of a system to actively optimize
Cost Reduction	To reduce the development cost of SOA, such as time
IT/Business Alignment	In which information technology (IT) is a dynamic state where the business organization is able to effectively use IT to achieve business objectives.
Business Quality	Ability to provide quality of service in the SOA system.
Business Optimization	Ability to be able to spread business processes from the organization.
Feasibility study	Feasible study to actual system.
Requirements elicitation and analyse	In this phase, engineers should contact the customers and end users to provide system domain, system performance, services, system barriers etc. to collect information. Requirements and analysis need to be done with different role of different stockholders.
Requirement specification	The goal of requirement specified easily.
Requirement validation	To validate the actual requirements
Modifiability	Modification is the ability to make changes in a system quickly and cost effectively. Any change made easily by the user
Performance	Performance in the SOA context is usually measured by the average case response time or throughput.
Testability	Test the valid requirements.
Query response time	Sending feedback to user and getting response.
Security	Cloud security ,web security
Cost	Cost total required process.

1.3 Survey done on Approx. 100 papers, publication dates ranging from 2009 to 2017

Before 2009 the research in FR and NFR was focused on the necessity and prioritization of the functional and non-functional requirements. The requirements were first standardized by, International Function Point Users Group , Which captured through fourteen general system characteristics the critical implementation features of an application. IFPUG used The Function point analysis (FPA) method to select the most critical features. In 2009 Bhardwaj et al. [2] suggested adding more Non- functional requirements such as Functionality, Reliability, Usability, Efficiency, Maintainability, Portability etc. that till date have not part of the FPA estimation method. The authors also discussed some NFRs and tried to determine an effect for each of them.

During the same year Christopher Burgess et al.[3] The focus is on determining the degree on which a particular software meets the system requirements. In the paper the authors presented a Amendment of NFR framework which allows for the discovery of a set of system capabilities that best satisfies a given set of non-functional requirements. This proposed technique presents an adaptation of the softgel mutual dependency graph, which reflects the softguol mutual dependency rule graph, in which labels can be continuously promoted. The proposed technology facilitates the use of optimization algorithms to determine the best set of low-level operating softgolds, which best satisfies the highest levels of NFR softgongs. The proposed method also offers the ability to include both qualitative and quantitative information.

Carlo Ghezzi et al. [4] Service took a different approach to the applications provided by the integrators and focused on non-functional requirements; That is, the software that provides services by typing the services is independently developed, managed and developed by other service providers.In particular, they insist on potentially expressed needs such as probability or performance. It also shows an integrated approach-a method and its support tool - which argues for satisfactory needs because the system is dynamically developed. The approach relies on run-time monitoring and uses the data collected by the investigation to determine whether the behavior of the open environment in which the application is located, such as the usage profile or the external services associated with the application at the beginning, in the beginning Show distracted notions and may cause application failure.

M. Kassab et al. [5] worked on NFR classification standardization. It has been suggested that normal communication is necessary to enable effective communication and enable integration of NFR related research activities within the RE community. In his paper, they tried to realize the common foundation by developing an ontology-based approach, which is enough to consider NFR and their relationships for projects first in software development and throughout the life cycle.

During 2009 some authors like Breno Lisi Romano et al. [7] suggested the development of an automated tool to perform the navigability testing of FR and NFR, and use them in real life scenarios.

To detect the effect of non-functional requirements on the system design process Anargyros Tsadimas et al. [8] Propose a model-based approach to emphasizing non-functional needs. During the architecture design, to facilitate designers to effectively define and handle designers, each of them is proposed to focus on different design issues and meet different types of needs. A continuous requirement model is defined how non-functional requirements are related to the system components that make up the overall system architecture. SysML has been adopted as modeling language, because this requirement enables the definition and can be formally expanded.

In 2010, features of non-functional requirements are so sophisticated that there are many topics that should be more clearly defined. To address one of the challenging topics, proposal for a systematic software development process to support the management of daagegen [9] non-functional requirements. There are six steps in the process, each of which is made up of detailed activities. Using the proposed process, non-functional requirements can be more effective and systemally managed and modeled.

David Ameller et al. [10] Report the current state of model-based driven development (MDD) approach in relation to NFR and comment that, in general, NFR is not addressed in MDD methods and procedures, and the effects of this situation are discussed. After this, formulate a general framework that integrates the NDR into the root of the MDD process and provides a detailed comparison between all the MDD approaches. Finally, identify some research issues related to this framework.

Component Based Software Development (CBD) has been identified as a skilled and effective technique for developing complex software systems. Many component selection techniques already address functional or non-functional requirements (NFRs), however, NFRs are hard to be satisfied because most are system-wide properties and can not be connected directly with components. There is no technology in dealing with cases of NFR sharing properties in system performance groups. Fernando Mancilla et al. [11] Current techniques are proposed to combine Costume and Azimut +, where cost group NFR-sharing capabilities and Azimut + generate a combination of candidate components for each group.

Sam Supakkul et al. [12] Presenting an approach to look at the NFR pattern, in which to achieve the goal of the purpose of aiming to acquire the knowledge of NFR, the problem pattern, the problem pattern, to prevent obstacles in achieving the goal pattern or goals Alternative Solution Patterns and Selection Patterns For Prevention To Choose Between Options In The Ideas Of Your Side Effects. In this approach, not only are individual NFR patterns but also the relationships between them are imagined, including expertise, structure, and urgency. An emphasis is placed on visualization support framework and model based tools, which capture, organize, implement and enforce integrity barriers on the basis of model amortization rules, to continually recapture the captured NFR models during each reuse. Is applied for.

In 2011 YOUSRA ODEH et al. [13] Present a new framework to classify non-functional requirements in connection with software engineering services and service-oriented systems. In addition, this new classification service is expected to make significant contribution in facilitating NFR identification and specification for engineering and service-oriented software engineering.

David Villegas et al. [14] Discuss the design and implementation of an infrastructure as a service cloud manager, that the non-functional requirements set during the required analysis phase can be mapped to the properties of a group of virtual devices running the application. During various development stages discussion management system ensures that quality service required to be maintained during the execution and to be considered.

Rolan Abdukalykov et al. [15] It is also a different approach and suggests a novel effort estimation method that can be used in the initial steps of software development projects. The proposed method initially clusters historical data from previous projects into separate problem domains, and domain-specific efforts generate estimation models, each involves the effect of NFR on effort by a fairly measured set of nominal characteristics is Reduce the complexity of these models using the feature subset selection algorithm.

Timo Frank et al. [16] Based on this, the non-functional requirements derived from the information science point-of-view are made to suit the specifications of automation techniques, based on which critical challenges are identified for the development of the distributed automation system.

IT convergence based applications rely on the integration of different technologies to provide different functionality. Subramaniam Ganesh et al. [17] It represents the important NFR framework related to IT convergence and through an example discusses the way to measure NFR reliability for IT infrastructure.

Aldrin Freddie Jaramillo presents an offer for appreciation of NFR from the business process model present in organizations [18]. The main contribution of the approach is to assist engineers with requirements in the systematic NFR alignment process using the heuristics and templates designed for this purpose, and organizational knowledge present in the form of business process model.

Syed Pajajdeh [19] model the proposed method system in the abstraction of three different levels. The characteristics and configurations of the components are considered in the lower layer of modeling. The effect of the configuration of components on the NFA of System Nodes is considered at the second level of the model. The NFA interconnection and NFA effects of the equipment on the system's total NFA are studied in the top layer. The analytical hierarchy process is suggested to determine degree of relative importance of various NFAs in different levels of system modeling. After modeling, an optimization process tries to find the best components and their settings to achieve optimal measures for the NFA of PHS.Comprehensive Healthcare Systems (PHS) missions are important systems that require high quality support for non functional features.

Jawaria Sadiq et al. [20] The goal of research is to improve quality in service-oriented applications by improving engineering requirements and quality requirements. Idea Quantification Mechanism, which covers service development from consumer perspective and is capable of returning to better quality requirements management in service oriented

Bo v et al. [22] makes a meaningful effort to inspect the implementation strategies of non-functional requirements in a potentially viable manner, and achieves possible results for the status of each alert. The contribution of work is to give a clear justification about whether there is a proper implementation strategy for many non-functional requirements so that they can be guaranteed a specific saturated condition, and if so, how big is the probability.

In 2012, Hannah Becha et al. Information about non-functional properties (NFPs) is seldom described in service-oriented architecture (SOA) services. [23]

Atepha Khosravi et al. [24] Explain the vitality of RE (requiring engineering) in the Service-Oriented Architecture (SOA) and offer guidelines for implementing RE in a service-oriented environment. The main point of SOA is to ensure that services can be developed by using the services provided by the consumer service providers to develop their applications. But the important point here is that the services should be accurate and according to the needs of consumers. Services should be reusable so that the development of new applications will be faster and its cost will be reduced. Therefore it is necessary to specify the requirements accurately and fully, and the need for engineering required emerges here.

The lack of a global perspective, which collects both aspects from time to time by analyzing requirements for conceptual design, because many DW projects fail to meet their goals in meeting users' needs and expectations.Mohammad EI Mohjir et al. [25] Will present the methods adopted for achieving this goal and by integrating three additional aspects in the growth, analysis analysis workflow, i.e. organizational decision making and business process ideas.

Problem Frames (PF) approach is a potential tool for classifying, analyzing, and structuring software development problems. However, there is not a practical way to catch NFR in the current stage.Bin yin et al. [26] System analysts have to expand the PF approach to help systematically capture non-functional requirements within the PF approach. A new problem model and a model-guided process have been presented. According to that process, NFR specifications can be obtained.

In the past, some empirical studies have been conducted, but none in the perspective of software architects, despite the great effect on NFR's daily architects practices. David Emler et al. [27] Depending on the architecture of the software, some findings of an empirical study presents. It addresses such questions: those who decide NFR, what kind of NFR architectures are important, how NFR documents are, and how NFR is valid.

System designers have to adhere to a large number of functional and non-functional requirements for large software projects, which make complex engineering work for software development. If these requirements change during the development process, complexity also increases. Antonina Danylenko et al. [28] Recommendations recommend system based on context-conscious structure so that the system designers are able to automate decision regarding efficiency of non-functional requirements, such as performance, and focus on the design of the basic functionality of the system Can go.

The result of the increase in automation and therefore the increasing complexity of automation functions requires the use of distributed control architecture. It shows a support for developers in engineering of such distributed automation systems. For this reason, this letter focuses on the design support of the automation system distributed using the design pattern. Important aspects are considering non-functional requirements in design patterns and the integration of design patterns in the engineering workflow .Karen Eckert et al. [29] presents a design pattern template that supports developers with predefined automation operations that are assigned to system functions and take proper non-functional requirements in account, in selecting appropriate distribution of automation tasks.

Weim M Fareed et al. [30] Model FR, NFR, and three fundamental agile artifacts are proposed for their possible solutions model in a visual environment. First, the FR is modeled through Agile Use Case (AUC). Second, NFR is modeled through Agile Lose Case (ALC). Third, NFR potential solutions are modeled through Agile Selective Cases (ACC).AUC is the new proposed hybrid of use cases and tight user stories. The ALC is offered a weakly defined agile NFR. ACC is offered a potential solution (operation) for ALC. Three artifacts are added to a visual framework to promote the tight modeling of NFR (mainly) and how they are connected to the FR. Artifacts are the building blocks of another broad framework for NFR modeling in agile software development processes.

Yi Liu et al. [31] A pattern based approach is proposed to design NFR, and integrates design results into existing functional UML models. NFR-specific patterns that reflect the general design of NFR solutions are proposed to reuse the design knowledge, and the aspects are automatically generated to integrate into the existing functional model.The benefits of this approach include filling gap between NFR analysis and implementation, using reuse of NFR design experience by pattern. Finally, to consider both functional requirements and non-functional requirements, one can get a more comprehensive design model.

The agreement on non-functional requirements between the customer and the supplier is important for a successful IT solution distribution project. In an ideal world, stakeholders and architects collaborate to achieve their common goals in the win-win situation. In a commercial setting, however, a prominent feature often introduces powerful forces from outside the technical field.This feature is a customer / supplier relationship, usually forming bidding rules or delivery contracts. Formal Customer / Supplier Relationships often put serious limitations on information exchange between stakeholders and architects. Eltjo R Poort et al. [32] Find out the impact of such borders on the interaction of architectural design with the quantities of non-functional requirements, and explore many ways to deal with them.

Services Oriented Computing is very popular in current product lifecycle management solutions. Customer functional and non-functional requirements should be included in a service oriented design. Clotilde Rohleder [33] There is an approach proposed on how to represent non-functional requirements in a deliberate service statement.

Mehrad Sadatmand et al. [34] Focus on these needs and identify what information about NFR is necessary to compare business closed analysis and design model and using the UML profiling method to annotate model elements with necessary information. Explain the approach to incorporate this information into the system model. Then calculate the satisfaction values of NFR using the model change technique.

Yasuhiro Saito et al. [35] Offers a simple assessment model of NFR involved in RFP, primarily focusing on user maintenance and operational issues. This model includes weight for NFR categories, NFR matrix, description level grading and for each NFR.As a case study, the proposed model evaluates RFP of 29 projects. As a result, it was confirmed that the model can identify poor written NF aspects in RFP, which requires finishing before asking the developer company for the proposal.

Norbert Sigmund et al. [36] Advertise the problem of achieving the interaction of non-functional requirements among many interactive systems using the real world scenario. Also, show an approach to finding optimal forms for many systems, which reduces the calculation effort through a phased configuration process.

Vikas Bajpey et al. [37] Give details on non-functional requirements and its importance in different areas. Also check the effect of working for non-functional requirements that lead to the discovery of new functional requirements. After the coding phase in the Software Development Life Cycle (SDLC), during the process of expressing non-functional requirements, software analysts have created new unexpected functional requirements and which further create a vicious cycle.

In 2013, the focus was changed on new technologies like cloud and wireless sensor networks. The authors began researching the FR and NFR to support the upcoming technologies.

Diana Carolina Barreto Arias [39] has played a leading role in the development of FR and NFR for these new technologies. The authors offered an architecture that supports the process of deploying applications in cloud providers using high level information based on non-functional requirements. Compared to experiments with prototypes on a small scale, our selection is based on a value based on non-functional requirements compared to the process. The results show that our architecture makes a lot of improvement in the process.

Ileana Iankoulova et al. [38] Cloud computing provides a comprehensive and structured overview of security requirements and solutions. We reviewed a systematic review and security requirements identified from previous publications in nine sub-areas: Access Control, Attacks / Harm Detection, Non-Disclaimer, Integrity, Security Audit, Physical Security, Privacy, Recovery, and Prosecution.It has been found that (i) the sub-areas that have been searched for at least are non-denial, physical security, recovery and prosecution, and (ii) access control, integrity and audit are the most searched sub-areas.

Sameer Bariani [40] proposes model-driven architecture (MDA) approach to improve verification of wireless sensor networks (WSN) properties. This approach connects the benefits of System Modeling Language (SISML) and Modelica language, which promotes reusability and improves the development process.

Cloud computing is an option for provision of resources, which can be done using the provider's own infrastructure or the infrastructure of one or more public clouds, or even a combination of both. This enables more flexible / elastic usage of resources, but the specification does not solve the problem. Rafael Gomes et al. [41] Uses models at runtime to facilitate the specification of non-functional requirements and resources, with the aim of facilitating dynamic support for application execution in cloud computing environments with shared resources.

In practice, it happens that non-functional aspects are often ignored while focusing on the system's performance. Many systems have failed because of negligence of non-functional requirements. To study the effect of non-functional requirements on the development of needs, Kiran Khair et al. [42] During the software development, discusses various approaches available in the literature to represent non-functional requirements. To study the effect of the non-functional requirements on the development of needs, three issues were focused on: different ideas on non-functional requirements, representation of non-functional requirements and dealing with non-functional requirements.

Davy Marieza et al. [43] In relation to NFR's relative characteristic, NFR presents a novel structure to manage conflicts. By applying an experimental approach, quantitative proofs of NFR conflicts will be obtained and modeled. In order to generate quantitative evidence, NFR metrics and measures will be used in experiments as parameters. This evidence can allow developers to identify and argue about NFR conflicts. Provide an example of how this framework can be implemented.

It is well known that NFR has a major impact on the overall cost and timing of the system development process, as they often describe cross-cutting concerns. To improve software development support, an automated analysis of SRS documents is required for various NFR types. Abderahman Rashwan et al. [44] Two significant contributions towards this goal are included in the work: (1) A new gold standard corpus that contains annotations for various NFR types, depending on the requirement science; And (2) A Support Vector Machine (SVM) classifies classified automatically to classify the requirements in different autology classes. The results obtained from two different SRS corporations show the effectiveness of the approach.

John Slunks et al. [45] Check which document types (data usage agreement, manual, regulations, requests for proposals, requirement specifications and user manual) include NFR, in which NFR classified 14 NFR categories (such as capacity, reliability and security) has gone. Measure how well we can identify and classify the NFR statement within these documents. In each of the evaluated documents, NFRs were found. Using a word vector representation of NFR, a support vector machine algorithm effectively demonstrated twice as much compared to the same input for a multi-functional unsightly bayous classification.

Ajit Tiwari et al. [46] Provide middleware between session management scenarios to provide QoS by feeling the flow of data packets between client side applications and application servers. It helps in obtaining reliability and scalability between services.

Anton V. Uzunov et al. [47] propose a form of attribute that we call architectural decomposition, and present a multi-level conceptual framework for dismantling distributed software architecture. Using the framework to dismantle the architecture, through complementary analysis processes, can help in determining NFR at the architectural level.Describe each level of the structure in return, offer a supplemental analysis process for safety based on threat modeling, as well as the process of using the framework, and demonstrate the usefulness of our approach through real life example Do

Distributed Architecture

In order to make better software, NFR should be considered as the criteria for design decisions. However, different NFR software can generate different criteria on the implementation strategies of the tasks. In order to satisfy NFR as satisfying as possible, a trade-off analysis is required to obtain the optimum plan during the design decision.By focusing on the NFR which can be quantitatively assigned, bin Yin et al. [48] There is an approach proposed to find such optimal solutions to help in making better decisions. This approach assumes NFR as constraints on the implementation strategies of software operations and selects the implementation strategies of the 0-1 programming problem. Then, 0-1 programming solver can be used to find the optimal solution.

Cloud computing is an option for provision of resources, which can be done using the provider's own infrastructure or the infrastructure of one or more public clouds, or even a combination of both. This enables more flexible / elastic usage of resources, but the specification does not solve the problem. Rafael Gomes et al. [49] Presenting an approach using the model on runtime to facilitate the specification of non-functional requirements and resources with the aim of facilitating dynamic support for application execution in cloud computing environments with shared resources.

In order to face problems of scalability and complexity of information system (IS) in 2014, the conceptual model should be able to understand the necessary requirements for its development. Beyond the idea of functional requirements, other important requirements have emerged: non-functional requirements to reflect complex situations that occur in the real world.Abdelhadi Boen et al. [50] Introduce an approach to the integration of non-functional requirements in the concept of Information Systems. The proposed approach is an approach based on service-based architecture (SOA), model-driven architecture (MDA), and automated changes to the model.

Lars Brübch et al. [51] One vision presents how both aspects can be brought to the same level and present an approach to increase component-based cloud applications with non-functional properties in the context of an integrated programming model. This allows the developers of such applications to formally declare those properties which can affect the application during runtime and document expectations about the non-functional aspects of the overall system.This can be used to ensure the health of the application and, if requirements are violated, then additional tasks such as deployment of operations can be done to reduce any degradation of their non-functional state. , Either by evaluating, selecting and displaying - automatically or by manual administrative work intervention.

Requirements Priority is recognized as an important but often neglected activity during the software development process. To achieve high-quality software systems, both the functional and non-functional requirements should be kept in mind during the priority process. Mohammed Tumhagh et al. [52] Propose an approach using Hybrid Assessment Method (HAM) to prioritize functional and non-functional requirements together.Evaluation of the effectiveness of the proposed approach has been done with the aim of comparing approaches with state-of-the-art approach to the analytical hierarchy process (AHP) through an experiment.

Software product line and component-based software engineering are two extended paradigms for the development of critical and complex systems. Therefore, supplemental powers appear in these two patterns. Therefore, significant benefits can be obtained from their integration.Regardless of the fact that non-functional requirements (such as security, performance and reliability) are important to the software system, and they should be kept in mind at the initial stage of software development life cycle and clearly specified with functional requirements. needed; These NFRs are ignored in software product line and component-based software development processes.Fatima Zahar Hammani et al. [53] Discuss some recent approaches in the field and offer a new extension for software product line processes which support component-based approach, and integrates the non-functional requirement at the stage of domain requirements.

Needs analysis is considered the most important step for the development of quality software because errors are likely to crawl in design and implementation phase due to poor and inadequate need analysis. These errors are considered to be the most expensive for time consuming and repair. Thus, the requirement analysis phase, which determines the success or failure of a software project.Kiran Khair et al. 54] It has been observed that while focusing on software performance, non-functional properties are often ignored. Many software systems have failed due to negligence of non-functional requirements. Therefore, it is necessary to measure the level of satisfaction level of non-functional requirement during the software development process.

Dependency between non-functional requirements (NFR) is one of the major issues to handle quality software. These dependency relationships are due to conflict, although there are links where an NFR helps in ensuring another NFR. An interdependent relation, not treated since the beginning of a software development, could sack the requirement in the later stages of development.In order to solve those issues, the interdependence between the NFR has been determined. Mirza Renu Tabasam et al. [55] A framework has been proposed to deal with interconnected NFR from the initial steps of the development project. The proposed NFR interdependence frame keeps the NFR with the functional requirements with which they are connected, which help in designing the system without leaving NFR for post-development stages.

Some authors also used NFR to finalize the requirements of various ICT based projects. Mujtaba b Ila et al. [56] An approach is proposed that will support the education stakeholders in choosing the most suitable ICT infrastructure architecture, which will conform to their needs based on the non-functional requirements of the proposed system (NFR).

Data Warehouse (DW) architecture, which are in the center of many new generation information systems, designed to give decision-makers and professional analysts an intuitive and executable medium to easily detect and understand intelligent data from the heterogeneous database. has been done.Most of the ways to deal with DW projects focus on functional requirements related to data quality; Non-functional requirements like security, performance or operation are considered completely different.The lack of a global perspective, which collects both aspects from time-to-time with the analysis of requirements from conceptual design, is the reason that many DW projects have failed to meet their goals in satisfying users' needs and expectations Mohamed El Mohzer Et al. [57] Developing a framework that includes both functional and non-functional requirements throughout the process of data warehouse design.These requirements are achieved by integrating three additional aspects in the analysis workflow, i.e. organizational decision-making and business process ideas.

Model-driven development (MDD) in 2015 is no longer a novel development model. This research has matured from perspective and recent studies have shown its adoption in the industry. Still, some issues are making a challenge. David Emler et al. [58] In the industry, MDG has informed the planning and current situation of an empirical study about NFR management.

The use of model driven development (MDD) in the industry is increasing. When non-functional requirement (NFR) is not considered in development then metamodels, models and conversions should also be affected. Work can be complicated in MDD to define and maintain model change rules. Model Transformation (MTBE) approaches, for example, have been proposed to reduce the development of change rules.Joseba a Agr et al. [5 9] NTR appears to be applied in a heritage model change when an approach based on MTBE to achieve adaptation operations. The approach achieves semi-automatic model transformation using performance traceability data and model differences.

Agile software development needs special focus on NFR. Requirements elicitation, design, development, and verification are important activities of implementation software development. Agile has good procedures to meet the functional requirements caught in the form of user stories. However, the NFR emissions have not been given enough attention in Agile processes. In addition, there is a lack of logic and verification of NFR.Darshan Doma et al. [60] "NVR method: Authentication, logic, and verification are proposed in non-functional requirements, Agile processes." Current results show that the artifacts developed in this research can potentially help software development organizations to address NFR in the next Agil procedures.

Provides an integrated range of uninterrupted model-based development models, in which all software engineering steps are included. Non-functional requirements (NFRs) play an important role in software and system engineering, as reusability, but are often neglected in research and practice. It is still unclear how to integrate NFR into an uninterrupted model-based development.The overall study design involves a multi-step process to estimate an experienced established theory on specifying NFR to support uninterrupted modeling. Jonas Anchart et al. [61] Studies present the design and provide discussion about the initial results (i) received from the sample, and (ii) current issues related to the design. So far we have concluded that our overall study design seems suitable for long-lasting fantasy theory.

Effective software project management is essential to make accurate software effort estimates that normally remain a major challenge for the software engineering and software industry. Fadoua FELLIR et al. [62] Based on the combination of COSMIC and case-based logic, the initial software size and effort estimation method is proposed, which uses the individual requirement measurement as a solution to improve CBR's performance and increase the accuracy of estimates.This hybrid technique involves adjusting FR measure with the effect of NFR, with which they are connected. A new link requirement model has been proposed, in which potential relationships between FR and NFR are expressed. This combination will help efficiently involve in the process of measuring their relationships with NFR, and FR and throughout the life cycle of the software development project.

Joshua Goncalves et al. [63] On the basis of expanded non-functional requirement structure, offer a fully model-driven approach. This approach allows agents to get directly from the extended NFR-Framework model. Therefore, the developer only needs knowledge and expertise in this specific framework, which is quite easy to understand.This derivative agent can select the units of the model in relation to their hierarchy, and may be influenced by whatever can contribute; Choosing plans which show the highest benefit to the overall system This model driven approach focuses on operating within a dynamic environment, which is the first of its kind, for our knowledge. This approach is used experimentally.

Tetiana Hovaroshchenko [64] The known approaches to detect emerging properties of software systems were analyzed in terms of partnership for software quality model. Attempts were made to evaluate the part of non-functional requirements in the software quality model.

NFRs are systematically examined to determine aspects that can be harmonious or cause controversy so that the conflict should be detected as soon as possible. Haibo hoo et al. [65] The goal of the work is to expand the NFR Framework and the SoftGold Inter dependency graph (SIG) for NFR's modeling and analysis.A formal formal approach has been proposed for NFR interaction modeling, and core NFR autology has been developed in the autologie web language. The mechanism involved in NFR interaction with SIG models (through interdependence in softgalls) is investigated and formalized.

Aspect Oriented Programming (AOP) and Aspect-Oriented Modeling (AOM) have been proposed as developmental methods for handling non-functional requirements for embedded systems, which usually include cross-cutting concerns regarding a functional "application . [66] On the basis of observation that non-functional aspects are system-level Need to develop perceptive,Note that application-level AOP is not enough to address non-functional requirements (since it usually ignores execution platform barriers). Then offer a development process that recognizes AOM with two standard development approaches for embedded systems: model-based development and platform based designs. The proposed method is illustrated through an example with logical execution time requirements.

One of the main challenges in addressing non-functional requirements (NFR) in designing systems is to keep their mutual dependencies and mutual impact in mind. For this reason, they can not be considered in isolation and careful balancing and business should be established among them.It makes it difficult to choose design decisions and features that cause the satisfaction of all different NFRs in the system, which becomes even more difficult when the complexity of the system increases. Mehrad Sadatmand et al. [67] Opinional logic and decision support system offer an approach based on which helps to identify different design options that lead to high satisfaction of NFR in the system.

Ji Zou et al. [6 9] Stack overflow acquired the content of stack overflow discussions to find the main topics of discussions, used topic models, latent dutchal allocation (LDA), and the use of the wordlist to find relationships between discussions and NFR Done Hot and Unsolved NFR, focus on NFR's development and trends in their discussions.

In web service research, on the basis of functional requirements, a lot has been paid to provide methods and tools for automated structure of services. Ilyass EL KASSMI et al. [70] While creating the desired web service overall, the user offers a seamless way to dynamically knit the user's non-functional requirements with functional requirements. The approach proposal is based on the finite state automaton formalization and on the DIVISE (Discovery and Visual Interactive Web Service Engine) framework.

Arun B. Krishna et al. [71] The importance of developing the development method represents the importance of the functional requirement of display and memory in embedded system development in real time when the development method is agile with incremental growth. During the growth cycle, efficient methods of tracking these standards are also described.

Non-functional requirements (NFRs) are quality attributes that define how a software product will do its functions. They are important and important for the success of any software in the market because they are considered different from other software products that provide similar functionality. The Agile method has become popular in the last few years, and although it improves the process of software development, there are many limitations in analyzing its requirements.Neglecting non-functional requirements is one of the biggest limitations in Agile. It does not provide widely accepted techniques for expanding and managing non-functional requirements. Bahia M. Aljalabi et al. [72] At present, NFR summarizes the two current approaches used for analysis, and then it provides an advanced approach to better analysis of NFR.The new approach is better than the current ones because it connects their strengths and removes their weaknesses.

Exmatic Design Theory is a general framework for system design process based on mapping between multiple domains. In exemetic design theory, only functional requirements (system 'functions) are considered in the design process, because in the exematic design non-functional requirements (system' attributes) are not included in the mapping between requirements and design parameters.Mohammad A. Mrrok et al. [73] To include non-functional requirements in mapping between problem domain and solution domain, please provide an extended view of the system design process based on the Acmeatic Design Theory Framework.A systematic method of incorporating non-functional requirements has been introduced in the design process and extended design metrics have been introduced, in which the mapping between the full set of requirements (functional and non-functional) and related design parameters (physical domain) Includes.

Anas Mahmood et al. [74] Software offers an uncomfortable, computationally efficient and scalable approach to extracting and removing NFR systems. Based on the main assumptions of the cluster hypothesis and information theory, the proposed approach uses meaningful knowledge embedded in the text content of specifications to find, classify and find high-quality software quality hurdles imposed by the functional characteristics of the system.Three experimental systems are used to perform experimental analysis. The results show that the proposed approach can find software NFR with average accuracy of 73%, so that these NFRs can be traced to their applications with adequate accuracy levels for practical applications.

Agile Software Engineering has been a popular method for developing software fast and efficiently. However, this method often considers the agile requirements (FRS) due to the nature of agile software development and strongly neglects the non-functional requirements (NFR). Neglecting NFR has a negative impact on software products, resulting in poor quality and high cost to fix problems in later stages of software development.Richard R. Maiti et al. [75] Software requirements such as documents and images propose a study to effectively collect NFR metadata from artifacts. This will be accomplished by reducing false positives to include NFR in the initial stages of software requirements collected with FR.Apart from this, this study will use historical trends to predict additional NFR ignored by architects and can be included with FR in early stages of agile software development. Apart from this, NFR's priority is to use stakeholders as well as to provide quality software to software engineers using existing FRS methods.Research focuses on NFR especially during the early stages of agile software development, on pre-studies. The goal of the study is to improve the earlier studies of NFF to provide effective techniques for prioritizing NFR and predicting NFR during the early stages of priority development and the effects of NFR on the software development process.

In 2016, with the priority of non-functional requirements (NFR) modeling and logic is a research area that needs more attention. Louis H. Garcia Poker et al. [76] The performance of RE-PREF, an approach that supports NFR and modeling of their preferences, and the discovery of potential scenarios where poorly selected preferences can either miss runtime systems or suggest unnecessary optimization The system (SAS) which can reduce the behavior of adaptive behavior themselves.Specifically, we show how the RE-pref is used in the remote data mirroring (RDM) system. The model of NFR and its preferences are analyzed using the dynamic decision network (DDN) and the Bayesian wonder.

Rashmi Fallnikar [77] has suggested a new framework that supports the process of deploying applications in cloud providers compared to the non-functional requirements (NFR), mainly in terms of high standard requirements. Experiments have already been organized with conclusive results, in which the barrier of the user is matched during the service selection process. The use aspect uses oriented paradigm (AOP), XML and graph changes.The relevance of the structure is illustrated using Remote Patient Monitoring (RPM) scenario. The results show that the model improves the process by identifying contradictory NFR. Based on this work, the next logical step is to expand it into cloud computing.

Unmasking non-functional requirements (NFRs) such as quality features, interface requirements, and software design barriers are important in finding architectural alternatives to software that begins with the initial design opinion. Pratek Singh et al. [78] According to their occurrence in many NFR classes, the NFR in the sub-classes with the help of rule-based classification techniques, using the thematic roles within the document, combines the need for automatic identification and classification of sentences.

Due to the large, odd audience of web applications, and due to its rapidly changing expectations, the overall requirement analysis approach is important to ensure the success of web engineering projects. In order to increase the quality of the resulting web applications, non-functional requirements (NFRs) should be considered.Satisfying them is a non-minor task, which depends on the decision about implementation of the functional requirements (FR) and prioritizing NFR for implementation. A satisfactory solution is a trade-off, where competitive NFR should be balanced.J. Zubcoff et al. [7 9] Framework how to evaluate and select the optimal configuration of the requirements for web application, Pareto efficiency can be a goal-oriented requirement analysis modeling supplement, whereas NFRs are balanced and maximized according to priority list.We thus focus on an empirical evaluation to verify whether our PETO method requirements improve the accuracy of design decisions during the analysis phase, and / or if it reduces the time required by the designers.

Harsimran Kaur et al. [80] In order to improve the modularity and complexity of the use case, attempts to offer an aspect-oriented approach with new artifacts for the necessary NFR model associated with any particular use case. These artifacts are the kernel to represent standard use case functionality, to reduce the risk associated with the usability situation, to apply the barrier to the suitability status and NFR Case. The proposal has been illustrated with an example and the metric has been prepared. The result acknowledges that the proposal is compared to the proposed prevailing views in the past.

Joel Carvalho et al [81] A Web Utility and Accessibility Test Automation Framework presents that can be applied to other types of non-functional requirements.

Jonas Anchart et al. [82] An experienced viewpoint examines this approach and aims to increase our understanding of NFR's nature by addressing system properties. Report on the 530 NFR classification extracted from the specifications of 11 industrial requirements and analyze the extent to which the NFR system describes the behavior. Our results show that most \ non-functional "requirements are not non-functional because they describe the behavior of a system.As a result, argue that many so-called NFRs can be handled as functional requirements.

Integrating various non-functional requirements in the structure of web services is a complex task and can lead to unwanted conversations that negatively affect the quality of service and the expected results. Ilyass EL KASSMI et al. [83] In view of their mutual dependencies, modeling and integrating non-functional requirements offer a FIND-STATE automata-based approach. The UPPAAL Model Testing Tool is used to legalize the formalities.Non-functional requirements were applied to our approach to safety in this work, and proposing a health care case study for experiments.

The architecture of a software system is the result of the succession of architectural design decisions. This type of decision allows the software to choose architectural options and software components when demand for a stakeholder is demanded. However, there is no framework that organizes appropriate selection of software components using architectural tactics.Gaston Mercuus et al. [84] Describes the selection of components using architectural tactics (compact) to evaluate off-the-shelf components from non-functional requirements (NFR). In order to suggest strategies according to NFR and a collaborative component search in compact, there is a scenario / strategy schema based on utility trees.

In the development of a new software solution, users and developers spend most of their efforts to model the functional requirements. Non-functional requirements (NFRs) are considered as the requirement of second class, are ignored by the end of the development cycle. They are often hidden, shaded and therefore, often neglected or forgotten. They are neglected because NFR is often difficult to model, develop and test.NFR becomes even more important in secure software design. Secure software design requires the modeling of privacy and integrity of the data passed through the software system with the availability requirements of the software system. Aspen Olmsted [85] focuses on modeling NFR and converting transformations into source code from UML model.

In 2017, continuous integration (CI) implies that a complete developer team works together on the main line of a software project. CI systems automate the creation of a software. Sometimes a developer checks in code, which breaks the build. A broken construction itself can not be a problem, but it has the ability to interfere with colleagues, so it affects the team's performance.Klérisson VR Paixão et al. [86] Important differences were found between NFR-related construction conditions. Thus, with the focus on new approaches to preventing CI failures, equipment can be proposed to improve CI, especially for the production of efficiency and utility.

Agil practices are gaining popularity over the speed of electricity in the software industry. When accepting major agile methods like scram and extreme programming (XP), customer requirements and changes in response, help to quickly deliver the system that fulfills functional requirements (FRS).However, the non-functional requirements (NFR) have been largely ignored or introduced in the software development cycle on a large scale and have been advocated for many project failures. Recently, changes in the side of the cloud and emphasis on large data in the software industry, such as NFR, have become more important than ever.Vaibhav Sachdev et al. [87] Using Scrum proposes a novel approach to handling security and performance for big data and projects related to the cloud. An industrial case study conducted in a period of 9 months reveals that the approach personally helps in dealing with conflicts between safety and performance requirements as well as in a tight method.

Mohamed Hamdi Irwan Hamza et al. [88] found that there is a lack of work, which provides a method on how to create matrix for evaluation of SOA adoption, focusing on both IT and business benefits. Therefore, the purpose of this study is to provide a method that can be used to create a cross-assessment matrix focused on SOA adoption IT and business benefits.In order to provide evaluation criteria for evaluation of SOA adoption, this study first determines IT and business benefit characteristics and sub-specialties. This study then adapted the Kano model to create cross-assessment matrix between IT and business benefits.

Conflicts between non-functional requirements (NFR) mean that achieving a requirement can have an impact on others. This can happen when a particular process (which is called "implementing softgold" in the literature) supports the first feature but creates difficulty for another. At present, the identity of the dispute uses catalogs to indicate NFR.Renara Maya Carvalho [8 9] proposes a list of disputes between the UBCump and the NOF of IOT applications, which consider their operating softballs. In addition to helping to create this catalog, this thesis uses the executable model to consider the design of softgöls operating in design time.

Zijad Kurtanovi´ et al. [90] Raise the second RE 17 Data Challenge: Identification of the type of requirements using the quality quality provided (NFR) "Datasets. We studied that we learned the functional machine with functional (FR) and non-functional datasets (NFR) can automatically classify the requirements.Apart from this, it was evaluated how accurately we can identify various types of NFRs, especially in the applicability, security, operations and performance requirements. We developed and evaluated meta-data, vocabulary, and sentence creative features to develop a supervised machine learning approach.

Umang Garg et al. [9 1] By presenting some existing techniques, present an approach that will help software engineers to prioritize the needs.

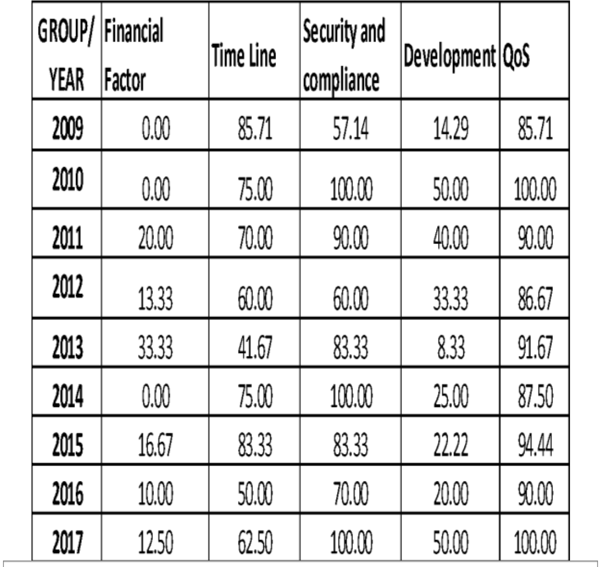
Requirements Engineering software plays an important role during the development process. Many actions have indicated that non-functional requirements (NFRs) are currently more important than functional requirements. NFR can be very complex to understand due to its diversity and subjective nature. NFR Framework has been proposed to fill the existing few intervals for NFR elicitation and modeling facility.Rodrigo Veleda et al. [9 2] Introduce a tool to help you find the stored knowledge to give solutions to implement quality requirements. The initial search mechanism is provided in this device to identify the possible solutions of NFR and to identify the other solutions and / or other related results of NFR.

It is difficult for customers to select adequate cloud providers who meet their needs, because the number of cloud offerings increases rapidly. Thus, many tasks focus on the design of cloud brokers. Unfortunately, most of them do not consider the exact safety requirements of customers. Asma Guesmi et al. [93] Depending on the functional and non-functional requirements, including security requirements, a method defined for placing services in a multi-provider cloud environment is proposed.

1.4 Outcome of survey

In this paper, all the NFRs proposed during last ten years have been discussed and on the basis of literature published related to the concerned NFRs a detailing has been done. This paper shows a timeline that how different NFRs has been used and evolved based on advent of other technologies. It is evident that change in technologies used in software development impacted the NFRs a lot. The data collected during this survey has been recorded and shown in the table and chart presented in the next section. Table 1 shows the percentage of different NFRs discussed in research literature during the subsequent years. The same is shown graphically in Figure 1.

Table 2 NFR focus record percentage wise Yearly



GROUP/ YEAR	Financial Factor	Time Line	Security and compliance	Development	QoS
2009	0.00	85.71	57.14	14.29	85.71
2010	0.00	75.00	100.00	50.00	100.00
2011	20.00	70.00	90.00	40.00	90.00
2012	13.33	60.00	60.00	33.33	86.67
2013	33.33	41.67	83.33	8.33	91.67
2014	0.00	75.00	100.00	25.00	87.50
2015	16.67	83.33	83.33	22.22	94.44
2016	10.00	50.00	70.00	20.00	90.00
2017	12.50	62.50	100.00	50.00	100.00

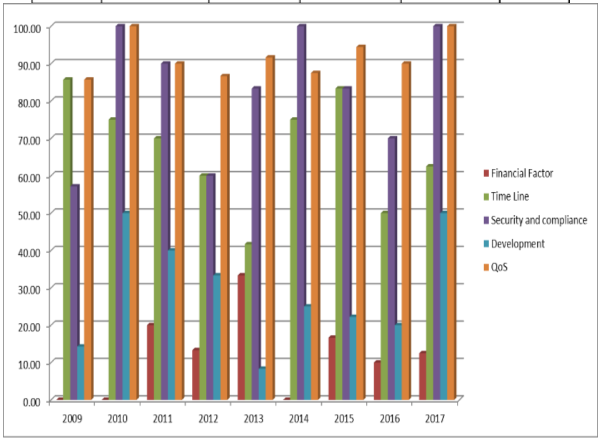


Fig. 2. NFR Chart from 2009 to 2017

1.5 Conclusion

In the conclusion of this paper author tries to find the quality factor terms more beneficial in comparison to financial factors. According to 2009 and 2010 years survey the financial factor find no graphically changes but in the same era quality factor rapidly change more times. So in this paper author finds the benefit of quality is more than financial quality factor and other factors like timeline and development is approximately low variation. Security factor is varies in different years which indicate that when security goes down then quality of service is also down. Financial factor is not an issue due to its variations and appearance is very less.

REFERENCES

(1) Aburub F., Odeh M. and Beeson I. Modeling Non-Functional Requirements for Business Process, Information and Software Technology Journal, vol. 49, no. 11-12, pp. 1162-1171, 2007. (journal)

(2) A.Keshav Bharadwaj, T.R. Gopalakrishnan Nair. Mapping General System Characteristics to Non-Functional Requirements, 2009 IEEE International Advance Computing Conference (IACC 2009) Patiala, India, 6-7 March 2009, (conference)

(3) Christopher Burgess, Aneesh Krishna, Li Jiang. Towards Optimising Non-Functional Requirements, 2009 Ninth International Conference on Quality Software (conference)

(4) Carlo Ghezzi and Giordano Tamburrelli. Reasoning on Non-Functional Requirements for Integrated Services, 2009 17th IEEE International Requirements Engineering Conference DOI 10.1109 (conference)

(5) M. Kassab, O. Ormandjieva. An Ontology Based approach to Non-Functional Requirements Conceptualization, 2009 Fourth International Conference on Software Engineering Advances DOI 10.1109 (conference)

(6) Quyen L. Nguyen. Non-Functional Requirements Analysis Modeling for Software Product Lines, MiSE'09, May 17-18, 2009, Vancouver, Canada (conference)

(7) Breno Lisi Romano¹, Gláucia Braga e Silva et al. New GenerationsSoftware Testing for Web-Applications Non-Functional Requirements, 2009 Sixth International Conference on Information Technology (conference)

(8) Anargyros Tsadimas, Mara Nikolaidou, Dimosthenis Anagnostopoulos. Handling non-functional requirements in Information System Architecture Design, 2009 Fourth International Conference on Software Engineering Advances DOI 10.1109/ICSEA.2009.18 (conference)

(9) Hyo Taeg Jung and Gil-Haeng Lee."A Systematic Software Development Process for Non-Functional Requirements 978-1-4244-9807-9/10/$26.00 ©2010 IEEE (conference)

(10) David Ameller, Xavier Franch. Dealing with Non-Functional Requirements in Model-Driven Development, 2010 18th IEEE International Requirements Engineering Conference (conference)

(11) Fernando Mancilla, Hernán Astudillo, Marcello Visconti. to address functional and non-functional requirements in software component selection Combining COSTUME and Azimut+, XXIX International Conference of the Chilean Computer Science Society (conference)

(12) Sam Supakkul and Lawrence Chung.Visualizing Non-Functional Requirements Patterns,978-1-4244-8765-3/10/$26.00 ©2010 IEEE (conference)

(13) YOUSRA ODEH 1 and MOHAMMED ODEH. A NEW CLASSIFICATION OF NON-FUNCTIONAL REQUIREMENTS FOR SERVICE-ORIENTED SOFTWARE ENINEERING, 2011 IEEE (conference)

(14) David Villegas and S. Masoud Sadjadi. Mapping Non-Functional Requirements to Cloud Applications, 2011 IEEE (conference)

(15) Rolan Abdukalykov, Ishrar Hussain, Mohamad Kassab, Olga Ormandjieva. Quantifying the Impact of Different Non-Functional Requirements and Problem Domains on Software Effort Estimation, 2011 Ninth International Conference on Software Engineering Research, Management and Applications (conference)

(16) Timo Frank1, Martin Merz1, Karin Eckert2, Thomas Hadlich3 et al. Dealing with non-functional requirements in Distributed Control Systems Engineering This work-in-progress paper was presented as part of the main technical program at IEEE ETFA'2011 978-1-4577-0018-7/11/$26.00 ©2011 IEEE (conference)

(17) Subramaniam Ganesan et al. Non-Functional Requirements for IT Convergence and Infrastructure, 2011 First ACIS/JNU International Conference on Computers, Networks, Systems, and Industrial Engineering, (conference)

(18) Aldrin Fredy Jaramillo, Non-functional requirements elicitation from business process models, 2012 IEEE (conference)

(19) Saeid Pashazadeh. Modeling Non Functional Requirements in Designing Middleware for Pervasive Healthcare System, 978-1-61284-832-7/11/$26.00 ©2011 IEEE (conference)

(20) Jawaria Sadiq Athar Mohsin Fahim Arif, Quantifying Non-functional Requirements in Service Oriented Development, 2011 Frontiers of Information Technology DOI 10.1109/FIT.2011.48 (conference)

(21) Saeed Ullah, Muzaffar Iq bal, Aamir Mehmood Khan. A Survey on Issues in Non-Functional Requirements Elicitation, 978-1-61284-941-6/11/$26.00 ©2011 IEEE (conference)

(22) Bo Wei, and Zhi Jin. Characterizing the Implementation of Software Non-Functional Requirements from Probabilistic Perspective, 2011 35th IEEE Annual Computer Software and Applications Conference (conference)

(23) Hanane Becha and Daniel Amyot. Non-Functional Properties in Service Oriented Architecture – A Consumer's Perspective JOURNAL OF SOFTWARE, VOL. 7, NO. 3, MARCH 2012 (journal)

(24) Atefeh Khosravi , Nasser Modiri. Requirement Engineering in Service-Oriented Architecture, IACSIT Press, Singapore DOI: 10.7763/IPCSIT.2012.V57.19 2012 International Conference on Networks and Information (ICNI 2012) IPCSIT vol. 57 (2012) © (2012) (conference)

(25) Mohammed EI Mohajir, Amal Latrache. Unifying and incorporating functional and non-functional requirements in data warehouse conceptual design, 978-1-4673-2725-1/12/$31.00 ©2012 IEEE (conference)

(26) Bin Yin and Zhi Jin. Extending the Problem Frames Approach for Capturing Non-functional Requirements, 2012 IEEE/ACIS 11th International Conference on Computer and Information Science (conference)

(27) David Ameller, Claudia Ayala, Jordi Cabot, Xavier Franch.How do Software Architects Consider Non-functional Requirements: An Exploratory Study,978-1-4673-2785-5/12/$31.00c 2012 IEEE RE 2012, Chicago, Illinois, USA, (conference)

(28) Antonina Danylenko and Welf Lowe. Context-Aware Recommender Systems for Non-functional Requirements, 978-1-4673-1759-7/12/$31.00c 2012 IEEE RSSE 2012, Zurich, Switzerland, (conference)

(29) Karin Eckert et al. Design Patterns for Distributed Automation Systems with Consideration of Non-Functional Requirements, 978-1-4673-4737-2/12/$31.00 ©2012 IEEE (conference)

(30) Weam M. Farid. Novel Lightweight Engineering Artifacts for Modeling Non-functional Requirements in Agile Processes, 978-1-4673-1375-9/12/$31.00 ©2012 IEEE (conference)

(31) Yi Liu et al. An Approach to Integrating Non-Functional Requirements into UML Design Models Based on NFR-Specific Patterns, 2012 12th International Conference on Quality Software (conference)

(32) Eltjo R. Poort et al. Issues Dealing with Non-Functional Requirements Across the Contractual Divide, 2012 Joint Working Conference on Software Architecture & 6th European Conference on Software Architecture (conference)

(33) Clotilde Rohleder. Non-Functional Requirements on Services – A Case Study, 2012 IEEE 36th International Conference on Computer Software and Applications Workshops Representing (conference)

(34) Mehrdad Saadatmand, Antonio Cicchetti, Mikael Sjodin. Toward Model-Based Trade-off Analysis of Non-Functional Requirements, 2012 38th Euromicro Conference on Software Engineering and Advanced Applications, (conference)

(35) Yasuhiro Saito. Evaluation of Non Functional Requirements in a Request For Proposal (RFP), 2012 Seventh International Conference on Software Process and Product Measurement DOI 10.1109/IWSM-MENSURA.2012.23 (conference)

(36) Norbert Siegmund, Maik Mory, Janet Feigenspan. Interoperability of Non-functional Requirements in Complex Systems, 978-1-4673-1853-2/12/$31.00c 2012 IEEE SEES 2012, Zurich, Switzerland (conference)

(37) Vikas Bajpai, Ravi Prakash Gorthi. On Non-Functional Requirements: A Survey, 2012 IEEE Students' Conference on Electrical, Electronics and Computer Science 978-1-4673-1515-9/12/$31.00 ©2012 IEEE (conference)

(38) Iliana Iankoulova. Cloud Computing Security Requirements: a Systematic Review, Date of Conference: 16-18 May 2012 IEEE Xplore: 16 July 2012 DOI: 10.1109/RCIS.2012.6240421 (conference)

(39) Diana Carolina Barreto Arias. Resource Provisioning in Clouds via Non-Functional Requirements, Department of Computing and Information Systems The University of Melbourne November, 2013(conference)

(40) Snmir berrani. Mapping SysMLto Modelica Validate Wireless Sensor Networks Non_Functional Requirements, 978-1-4799-1153-0/13/$26.00 ©2013 IEEE (conference)

(41) Raphael Gomes et al. On Modeling and Satisfaction of Non-Functional Requirements using Cloud Computing, 978-1-4799-3550-5/13/$31.00 ©2013 IEEE (conference)

(42) Kiran Khatter et al. Impact of Non-functional Requirements on Requirements Evolution, 2013 Sixth International Conference on Emerging Trends in Engineering and Technology 978-1-4799-2560-5/13 $31.00 © 2013 IEEE DOI 10.1109/ICETET.2013.15 61 2561-2 (conference)

(43) Dewi Mairiza, Didar Zowghi. Tools and Techniques • September 22-24, 2013, Budapest, Hungary SoMeT 2013 • 12th IEEE International Conference on Intelligent Software Methodologies, 978-1-4799-0421-1/13/$31.00 ©2013 IEEE (conference)

(44) Abderahman Rashwan Olga Ormandjieva Ren´e Witte. Ontology-Based Classification of Non-Functional Requirements in Software Specifications: A new Corpus and SVM-Based Classifier, 2013 IEEE 37th Annual Computer Software and Applications Conference (conference)

(45) John Slankas and Laurie Williams. Automated Extraction of Non-functional Requirements in Available Documentation, 978-1-4673-6271-9/13/$31.00 c 2013 IEEE NaturaLiSE 2013, San Francisco, CA, USA, (conference)

(46) Ajit Tewari, Shailendra Tripathi, Vikas Bajpai, Ravi P Gorthi. Role of Middleware to fulfillNon Functional Requirements (Reliability And QoS) in Multi-Session Pervasive Computing Application 978-1-4673-6190-3/13/$31.00 ©2013 IEEE (conference)

(47) Anton V. Uzunov, Katrina Falkner. Decomposing Distributed Software Architectures for the Determination and Incorporation of Security and Other Non-Functional Requirements, 2013 22nd Australian Conference on Software Engineering (conference)

(48) Bin Yin, Zhi Jin, Wei Zhang, Haiyan Zhao, Bo Wei. Finding Optimal Solution for Satisfying Non-Functional Requirements via 0-1 Programming, 2013 IEEE 37th Annual Computer Software and Applications Conference DOI 10.1109/COMPSAC.2013.69 (conference)

(49) Raphael Gomes et al. On Modeling and Satisfaction of Non-Functional Requirements using Cloud Computing 978-1-4799-3550-5/13/$31.00 ©2013 IEEE (conference)

(50) Abdelhadi BOUAIN. Integration of non-functional requirements in a service-oriented and model-driven approach, 978-1-4799-2393-9/14/$31.00 ©2014 IEEE (conference)

(51) Lars Braubach, Kai Jander and Alexander Pokahr. A Middleware for Managing Non-Functional Requirements in Cloud PaaS 2014 IEEE International Conference on Cloud and Autonomic Computing (conference)

(52) Mohammad Dabbagh et al. Application of Hybrid Assessment Method for Priority Assessment of Functional and Non-functional Requirements, 978-1-4799-4441-5/14/$31.00 ©2014 IEEE (conference)

(53) Fatima Zahra HAMMANI et al. Towards a Variable Non-Functional Requirements Integration for Component-Based Product Line A Generic Approach, 978-1-4799-4647-1/14/$31.00 ©2014 IEEE (conference)

(54) Kiran Khatter, Quantification of Non-functional Requirements, 978-1-4799-5173-4/14/$31.00 ©2014 IEEE (conference)

(55) Mirza Rehenuma Tabassum, Md. Saeed Siddik, Mohammad Shoyaib, Shah Mostafa Khaled. Determining Interdependency Among Non-functional Requirements to Reduce Conflict 3rd INTERNATIONAL CONFERENCE ON INFORMATICS, ELECTRONICS & VISION 2014 978-1-4799-5180-2/14/$31.00 ©2014 IEEE (conference)

(56) Mujtaba B Ila. Selecting an Effective Information and Communication Technology Architecture for an Education System based on Non-functional Requirements ©2014 IEEE (conference)

(57) Mohammed El Mohajir. TOWARDS A FRAMEWORK INCORPORATING FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS FOR DATAWAREHOUSE CONCEPTUAL DESIGN IADIS, International Journal on Computer Science and Information Systems Vol. 9, No. 1, pp. 43-54 ISSN: 1646-3692 (Journal)

(58) David Ameller, Xavier Franch, Cristina G´omez.Track Handling Non-functional Requirements in Model-Driven Development: An Ongoing Industrial Survey, 978-1-4673-6905-3/15/$31.00 c 2015 IEEE RE 2015, Ottawa, ON, Canada RE:Next! (conference)

(59) Joseba A. Agirre, Goiuria Sagardui and Leire Etxeberria. Evolving Legacy Model Transformations to Aggregate Non Functional Requirements of the Domain, MODELSWARD 2015- 3rd International Conference on Model-Driven Engineering Development (conference)

(60) Darshan Domah / Frank J. Mitropoulos. The NERV Methodology: A Lightweight Process for Addressing Non-functional Requirements in Agile Software Development, Proceedings of the IEEE SoutheastCon 2015, April 9 - 12, 2015 - Fort Lauderdale, Florida 978-1-4673-7300-5/15/$31.00 m015 IEEE (conference)

(61) Jonas Eckhardt, Daniel Méndez Fernández, Andreas Vogelsang. How to specify Non-functional Requirements to support seamless modeling? 978-1-4673-7899-4/15/$31.00 ©2015 IEEE (conference)

(62) Fadoua FELLIR et al.Analyzing the non-functional requirements to improve accuracy of software effort estimation through Case Based Reasoning, 78-1-4799-7560-0/15/$31 ©2015 IEEE (conference)

(63) Joshua Goncalves. Dynamic Non-Functional Requirements Based Model-Driven Agent Development, 2015 24th Australasian Software Engineering Conference DOI 10.1109/ASWEC.2015.24 (conference)

(64) Tetiana Hovorushchenko. COMPUTER SCIENCE & INFORMATION TECHNOLOGIES (CSIT'2015), 14-17 SEPTEMBER 2015, LVIV, UKRAINE (conference)

(65) Haibo Hu et al. Semantic modelling and automated reasoning of non-functional requirement conflicts in the context of softgoal interdependencies, ISSN 1751-8806 Received on 17th July 2014 Revised on 3rd November 2014 Accepted on 1st May 2015 doi: 10.1049/iet-sen.2014.0153 (Journal)

(66) Stefan Resmerita, Anton Poelzleitner, and Stefan Lukesch. Addressing Non-functional Requirements for Embedded Applications with Platform Based Aspect Design, 2015 IEEE 18th International Symposium on Real-Time Distributed Computing Workshops (conference)

(67) Mehrdad Saadatmand, Sahar Tahvili. A Fuzzy Decision Support Approach for Model-Based Tradeoff Analysis of Non-Functional Requirements, 2015 12th International Conference on Information Technology - New Generations DOI 10.1109/ITNG.2015.24 (conference)

(68) S. Sepúlveda, Member, IEEE, M. Bustamante, and A. Cravero. Identification of Non-Functional Requirements for Electronic Voting Systems: A Systematic Mapping, IEEE LATIN AMERICA TRANSACTIONS, VOL. 13, NO. 5, MAY 2015 (conference)

(69) Jie Zoub, Ling Xua,b, Weikang Guob, Meng Yanb, Dan Yangb, Xiaohong Zhanga. Which Non-functional Requirements do Developers Focus on? 2015 12th Working Conference on Mining Software Repositories DOI 10.1109/MSR.2015.60 (conference)

(70) Ilyass EL KASSMI, Zahi JARIR. 2015 International Conference on Protocol Engineering (ICPE) and International Conference on New Technologies of Distributed Systems (NTDS). 978-1-4673-9265-5/15/$31.00 ©2015 IEEE (conference)

(71) Arun B Krishna. Statistical analysis of memory and performance non functional requirements in real time embedded system development for agile methodology, 2015 International Conference on Industrial Instrumentation and Control (ICIC) College of Engineering Pune, India. May 28-30, 2015 978-1-4799-7165-7/15/$31.00 ©2015 IEEE (conference)

(72) Bahiya M.Aljallabi. Enhancement Approach for Non-Functional Requirements Analysis in Agile Environment, International Conference on Computing, Control, Networking, Electronics and Embedded Systems Engineering, 2015 978-1-4673-7869-7/15/$31.00 ©2015 IEEE (conference)

(73) Mohamed A. Mabrok et al. Including Non-Functional Requirements in the Axiomatic Design Process 978-1-4799-5927-3/15/$31.00 ©2015 IEEE (conference)

(74) Anas Mahmoud. An Information Theoretic Approach for Extracting and Tracing Non-functional Requirements, 978-1-4673-6905-3/15/$31.00 c 2015 IEEE RE 2015, Ottawa, ON, Canada (conference)

(75) Richard R. Maiti. Non-Functional Requirements Metadata During the Early Stages of Agile Software Development, Proceedings of the IEEE SoutheastCon 2015, April 9 - 12, 2015 - Fort Lauderdale, Florida Capturing, Eliciting, Predicting and Prioritizing (CEPP) 3718.-010-4 6©732-0713050 I-E5/E1E5 /$31.00 ©2015 IEEE(conference)

(76) Luis H. Garcia Paucar. RE-PREF: Support for REassessment of PREFerences of Non-functional Requirements for Better Decision-making in Self-adaptive Systems 2016 IEEE 24th International Requirements Engineering Conference DOI 10.1109/RE.2016.29 (conference)

(77) Rashmi Phalnikar. Validation of Non-Functional Requirements in Cloud Based Systems 2016 5th IEEE International Conference on Cloud Networking (conference)

(78) Prateek Singh, Deepali Singh, Ashish Sharma. Rule-Based System for Automated Classification of Non-Functional Requirements from Requirement Specifications, 2016 Intl. Conference on Advances in Computing, Communications and Informatics (ICACCI), Sept. 21-24, 2016, Jaipur, India 978-1-5090-2029-4/16/$31.00 @2016 IEEE (Conference)

(79) J. Zubcoff, I. Garrigós, S. Casteleyn, J. N. Mazón and J. A. Aguilar. Evaluating the use of Pareto Efficiency to Optimize Non-Functional Requirements Satisfaction in i* Modeling, IEEE LATIN AMERICA TRANSACTIONS, VOL. 14, NO. 1, JAN. 2016 (Conference)

(80) Harsimran Kaur. A Measure for Modelling Non-Functional Requirements using Extended Use Case, 978-9-3805-4421-2/16/$31.00_c 2016 IEEE (Conference)

(81) Joel Carvalho, Simão Melo de Sousa. Automated Analysis of Non-Functional Requirements for Web Applications, DOI: 10.1109/CISTI.2016.7521392. Conference: Conference: 2016 11th Iberian Conference on Information Systems and Technologies (CISTI) (Conference)

(82) Jonas Eckhardt, Andreas Vogelsang, and Daniel Méndez Fernández. Are "Non-functional" Requirements really Non-functional? 2016 IEEE/ACM 38th IEEE International Conference on Software Engineering DOI: http://dx.doi.org/10.1145/2884781.2884788 (Conference)

(83) Ilyass EL KASSMI and Zahi JARIR. Non-Functional Requirements Interdependencies in Web Service Composition, 978-1-5090-4926-4/16/$31.00 ©2016 IEEE (Conference)

(84) Gastón Márquez. Selecting components assemblies from non-functional requirements through tactics and scenarios 978-1-5090-3339-3/16/$31.00 _c 2016 IEEE (Conference)

(85) Aspen Olmsted. Secure Software Development through Non-Functional Requirements Modeling, International Conference on Information Society (i-Society 2016) 978-1-908320/61/2/$31.00 © 2016 IEEE (Conference)

(86) Klérisson V. R. Paixão∗, Crícia Z. Felício†, Fernanda M. Delfim∗ and Marcelo de A. Maia. On the Interplay between Non-Functional Requirements and Builds on Continuous Integration, 2017 IEEE/ACM 14th International Conference on Mining Software Repositories (MSR) DOI 10.1109/MSR.2017.33 (Conference)

(87) Vaibhav Sachdeva and Lawrence Chung. Handling Non-Functional Requirements For Big Data and IOT Projects in Scrum, 978-1-5090-3519-9/17/$31.00_c 2017 IEEE (Conference)

(88) Mohd Hamdi Irwan Hamzah, Fauziah Baharom and Haslina Mohd. A Service-Oriented Architecture Adoption Maturity Matrix using Kano Model: Cross Evaluation between IT and Business. e-ISSN: 2289-8131 Vol. 9 No. 2-12(Journal)

(89) Rainara Maia Carvalho. Dealing with Conflicts between Non-Functional Requirements of UbiComp and IoT Applications 2017 IEEE 25th International Requirements Engineering Conference DOI 10.1109/RE.2017.51 (Conference)

(90) Zijad Kurtanovi ET AL.Automatically Classifying Functional and Non-Functional Requirements Using Supervised Machine Learning, 2017 IEEE 25th International Requirements Engineering Conference (Conference)

(91) Umang Garg. Software Requirement Priortization based on Non-Functional Requirements, 978-1-5090-3519-9/17/$31.00c 2017 IEEE (conference)

(92) Rodrigo Veleda et al. Towards a Tool to Help Exploring Existing Non- Functional Requirements Solution Patterns 2017 IEEE 25th International Requirements Engineering Conference Workshops DOI 10.1109/REW.2017.49 (conference)

(93) Asma Guesmi, Patrice Clemente, Fr´ed´eric Loulergue and Pascal Berthom´e. Cloud Resources Placement based on Functional and Non-functional Requirements, (book)

First Author-

Raghu Veer Ph.D(P) from Dr. A.P.J. Abdul Kalam Technical University, Lucknow Assistant Professor Computer science and Engineering Department Babu Banarasi Das National Institute of Technology and Management, Lucknow India (Recognized by AICTE, Govt. of India & Affiliated to Dr. A.P.J. Abdul Kalam Technical University, Lucknow)

Second Author

Deepak Kumar Singh completed his doctorate from Indian Institute of Technology, Kanpur and has over Eight years' experience in the field of academics and IT industry. His core areas of expertise include Software Project management, Information Security, Machine Learning, IoT and Big Data Analytics. During his doctoral research tenure, he was nominated as a member of Indian delegation for SAARC-IAESTE Internship Program, Japan. Currently he is providing IT consultancy to several firms including UP Government. Dr. Deepak has played key roles several research projects, which include implementation of Learning Management Solution in several education institutions and two projects funded by Government of India. Also, he has been nominated by various forums and associations in their committees. He has presented more than fifteen papers in international journals and conferences and various seminars organized by professional bodies and industry associations worldwide. In addition to this he holds professional associations with IEEE Technical Committee, IEEE Young Professionals and Association of Knowledge Workers of Lucknow.