

---

## A versatile approach for ranking and modelling of non-functional requirements

---

Harsimran Kaur\* and Ashish Sharma

GLA University Mathura,  
Mathura, Uttar Pradesh 281406, India  
Email: harsimran.kaur\_csphd12@gla.ac.in  
Email: ashish.sharma@gla.ac.in  
\*Corresponding author

**Abstract:** To effectively encode domain knowledge of customers and implementation strategies of developers in critical computer based system (CBS), requirement engineering plays a significant role. For development of quality software, it is indeed necessary to specify both functional and non-functional requirements (NFRs). However, unlike functional requirements, NFRs are not given much importance and are often generically captured which results in missing out on quality parameters during CBS development. This lacuna can be taken care with inclusion of details regarding NFRs at initial stages of CBS development. NFRs can be addressed by use of interpretive structural modelling (ISM) and analytical hierarchy process (AHP) methods for identification of critical NFRs. Further, new artifacts have been introduced in use case diagram and reference model to document and validate the identified critical NFRs. Each proposed artefact in software requirement specification (SRS) document was initially analysed individually and then compared with other noticeable methodologies.

**Keywords:** non-functional requirements; NFRs; software requirement specification; SRS; use case; reference model; interpretive structural modelling; ISM; formal modelling.

**Reference** to this paper should be made as follows: Kaur, H. and Sharma, A. (2019) 'A versatile approach for ranking and modelling of non-functional requirements', *Int. J. Critical Computer-Based Systems*, Vol. 9, Nos. 1/2, pp.3–25.

**Biographical notes:** Harsimran Kaur received her MTech (CSE) and BTech (CSE) from the PTU Jalandhar. She obtained her PhD in the area of software engineering in the year 2018 from the GLA University Mathura. She has experience of eight years of teaching as an Assistant Professor. She has been teaching software engineering, java programming and analysis and design of algorithms.

Ashish Sharma is having a vast experience of around two decades in teaching and research. He received his PhD from the MNNIT, Allahabad and has published more than 40 papers in international journals and conferences. His research area includes software engineering, project management, artificial intelligence and distributed computing. He has guided more than 20 MTech and PhD thesis and currently guiding four PhD theses.

This paper is a revised and expanded version of a paper entitled 'A measure for modelling non-functional requirements using extended use case' presented at IEEE INDIACOM, New Delhi, 16–18 March 2016.

---

## 1 Introduction

Non-functional requirements (NFRs) are always considered as an important parameter to be indexed while writing in software requirement specification (SRS) document for critical computer-based system (CBS). But NFRs have been specified in SRS document either textually and sometimes informally by using UML diagrams particularly use case diagram. NFR's are merely mentioned as remarks which make it hard to organise the structure of NFR specifications. Numerous strategies and procedures have been proposed in past to enhance NFR identification and specification process. Yet at the same time for all goals and objectives, it is impractical to concentrate on each NFR during programming development as they are often interdependent on each other in contended ways. Hence, there is a need to settle for best possible compromise choices among the NFRs so that critical NFRs can be handled successfully with required detail and specified along with functional requirements while documentation in SRS document.

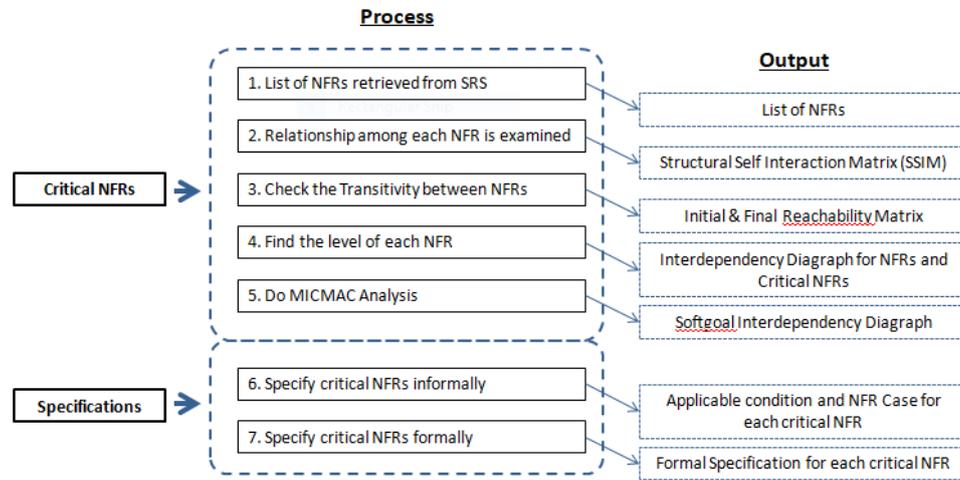
This article reflects on the following issues that have not been considered customary in leading studies:

- First, to identify critical NFRs and determining degree of mutual reliance between them so that software analysts can figure out such NFRs during the underlying periods of programming advancement.
- Second, to confine NFRs with the help of use case diagram so that it can address the changing needs of different classes of users.
- Third, to validate critical NFRs with the help of formal model.

The proposed approach to handle above issues consists of the following three modules:

- In order to identify critical NFRs, an approach using integration of interpretive structural modelling (ISM) and analytical hierarchy process (AHP) has been proposed. Main motive of approach is to bargain interdependencies between NFRs by generating softgoal interdependency digraph (SID) and MICMAC analysis to figure out critical NFRs. It will help the developers to finalise the NFRs which should remain in their priority list right from the analysis phase.
- Further, to specify these critical NFRs, standard use case diagram is extended with artefacts applicable condition and NFR case to model NFRs. The proposed extensions help to include levels of detail regarding NFRs necessary for the system engineers in system development.
- Finally, reference model is identified to integrate NFRs with formal model for validating the NFRs. It promotes correctness, viable, traceable and visible to different classes of users at an analysis.

The paper objective is to alter the traditional system of functional software engineering to the quality-based software engineering by specifying NFRs in much early phase of requirement gathering and analysis. In order to examine the credibility of proposed work, each proposed add-ons in SRS document was initially analysed individually and then later is compared with other built up and noticeable methodologies proposed in the past. The approach offered is shown in Figure 1.

**Figure 1** Model process (seven-step analysis) (see online version for colours)

The rest of article is structured as follows. Section 2 passes on the work carried out so far by a researcher which is related to NFR identification, specification and ranking. Section 3 discusses proposed an approach associated with integration of ISM and AHP, to figure out critical NFRs. Section 4 proposes the novel way of specifying identified critical NFRs in SRS both informally and formally. Section 5 examines the credibility of proposed work, each proposed add-ons in SRS document was initially analysed individually and then later is compared with other built up and noticeable methodologies proposed in the past. Cognitive dimension analysis is done for additional insight to the perceived usability of the extended version of SRS.

## 2 Related works

In literature there are number of informal definitions for NFR as summarised by Glinz (2007). According to IEEE (1993) “non functional requirement (NFR) – in software system engineering, 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.”

Boehm et al. (1976), McCall and Matsumoto (1980), Roman (1985), Grady and Caswell (1987), Sommerville (2006), ISO/IEC 25010 (2011) model, Glinz (2007), Mairiza et al. (2010), Chung et al. (2009), Fan and Liu (2010) and Jureta et al. (2006) addressed concerns to identify and classify the NFRs. But none of the model has been able to solve subjectivity associated with number of NFR’s and their interdependency. There has been very few researches on NFRs ranking in software systems but those are very specific to the particular software application. Some of ranking techniques for NFR are discussed here. Karlsson et al. (1998) concluded AHP have the potential to prioritise requirements. They used it in an industrial follow-up to further investigate its applicability. Liaskos et al. (2013) plugged AHP in the goal model by mapping every OR-decomposition in the goal model into a separate decision problem. They treat all non-

functional requirements as mandatory which is not practically possible. Elahi and Yu (2011) describe the quantitative approach requirements hierarchy approach (RHA), to determine and operate the consequences that NFRs have on a system without focusing on NFR interdependencies. Firesmith (2004) like many other authors (Yen and Tiao, 1997) enlisted various prioritisation techniques in his paper but did not prioritise mathematically. Kassab and Kilicay-Ergin (2015) provides a set of guidelines to transform NFRs framework into an AHP decision hierarchy. It directs practical solutions to rank the set choices for NFRs while considering their interdependencies. But there is need to check for inconsistency of the priorities calculated by AHP. Zhu et al. (2012) proposes a fuzzy qualitative and quantitative soft goal interdependency graphs (FQSIG) model for NFRs in Trustworthy Software and presents a tool based on MATLAB. It used the relation matrix algorithm that has ample with both negative and positive impact of NFR.

Thus, the literature to date lacks research on approach which can find critical NFR along with the interdependency between NFRs and can be applied on all systems. Undeniably, this is an important issue that needs to be significantly focused on. Among all the representation techniques (Aburub et al., 2007; Berenbach and Gall, 2006; Booch et al., 1999; Ernst et al., 2006; Jacobson and Cook, 2010; Jürjens, 2002; Warmer and Objecten, 2015; Liaskos et al., 2013; Lodderstedt et al., 2002; Mylopoulos et al., 1992; Pavlovski and Zou, 2008) the goal oriented approach considered as the first to represent NFRs in detail and later used by many researchers for their techniques like KAOS and NFR framework (Chung et al., 2009; Cysneiros et al., 2004). Few researchers tried to integrate NFR with FR (Aburub et al., 2007; Moreira et al., 2005; Rao and Gopichand, 2011; Supakkul and Chung, 2004; Tonu Subrina and Tahvildari, 2005). Researchers proposed extensions of use case and other diagrams like class diagram, activity diagram for documenting information systems (Fuentes-Fernández and Vallecillo, 2004; Zheng et al., 2010) but these extensions are application specific. NFR framework with UseCase is considered as one of the prominent work in this field which introduced the concept of soft goal to represent NFRs informally. But it increases the complexity of use case diagram. So, there is a need to extend the use case diagram beyond its current scope to address the changing needs of many different classes of users. Hence new artefacts are introduced in use case to provide complete representation of the overall system without increasing the complexity from user point of view.

It has been seen from the past failures of software that an informal specification does not contribute much while validating the NFRs. All the formal models in literature so far has validated only functional requirements (Gunter et al., 2000; Karlsson et al., 1998; Miller and Tribble, 2001). Hence there is a great need to have a formal model for NFR validation. Chung (2009) also discusses about need an incorporation of NFRs into other requirements models. Therefore reference model has been used along with use case model to promote correctness in development of NFRs.

### **3 Identifying critical NFR from SRS**

SRS is considered as the basic document for any project to start the development process. The thought is that, if the SRS is foundation of any software, can we not incorporate the elaborated details to NFR in SRS? Since as seen from the past failures that ignoring NFRs scrutiny has an exceedingly high payoff for the speculations, therefore it is

required to bring out a detail and organised analysis of NFRs for software to be developed. The first step toward this detail analysis is to identify the critical NFRs as discussed below.

### 3.1 List of NFRs retrieved from SRS

There ought to be group of cross-functional partners that assume diverse roles in programming advancement process for managing NFRs. In this paper partner (specialists) constitutes two academicians in the area of programming building and one programming designer from Software Company. Complete approach is discussed with the example of cafeteria ordering system (COS) that will allow process impact workers to request suppliers from the organisation cafeteria which are ordered online and to be delivered to specific requested destinations. NFRs taken from SRS for COS are tabled in Table 1. Critical NFRs recognised using SRS are then investigated by group of specialists to determine quantum relationship among NFRs.

**Table 1** Non-functional requirements for COS

<i>NFR</i>		<i>NFR requirements</i>	<i>Referred as</i>
Performance	PE-1	The system shall oblige 400 clients during the peak time usage window of 8:00 am to 10:00 am, with normal average session span of 8 minutes.	N1
	PE-2	All Web pages created by the system framework should be completely downloadable in close to 10 seconds over a 40 KBps modem association.	N2
	PE-3	Reactions to questions should not take more than 7 seconds to stack onto the screen after the client presents the inquiry	N3
	PE-4	The framework should show affirmation messages to clients inside 4 seconds after the client submits an initial information to the framework	N4
Security	SE-1	Network transactions which may be either financial of personal identifiable, must be encrypted per BR-33 standard.	N5
	SE-2	Clients will be required to sign in to the specially designed cafeteria ordering system for all operations except menu options.	N6
	SE-3	Benefactors should log in into the restricted computer system access policy as per the BR-35 standard.	N7
	SE-4	The system must privilege cafeteria staff individuals only who are listed approved menu managers to make or alter menus, per BR-24 standard.	N8
	SE-5	Just the users who have been authorised for home access to the corporate intranet may utilise the COS from non-organisation areas.	N9
	SE-7	The system should allow patrons to see just their own already placed orders while the order history of any other patron should be restricted from any unauthorised spoof.	N10

**Table 1** Non-functional requirements for COS (continued)

<i>NFR</i>	<i>NFR requirements</i>	<i>Referred as</i>
Robustness	On the off chance that the association between the client and the system terminates before a request being either confirmed or scratched off, the cafeteria ordering system must empower the user to recoup an incomplete request.	N11
Availability	The cafeteria ordering system should be accessible to clients on the corporate Intranet and to dial-in clients with an up-time of 99.9% for local time between 5:00 am to 12:00 in mid night and with compromised up time of about 95% for duration between 12:00 midnight till 5:00 am in morning.	N12

### 3.2 Relationship among each NFR

For strategic analysis of NFR's the relevant relationship is recognised amongst these NFRs and this leads to the accomplishment of self-connection grid (SSIM). Group of specialists chosen in Section 3.1.1 are then consulted in distinguishing the nature of relevant correlations among the NFR's. Two academicians with expertise in field of programming designing and one programming engineer with similar expertise from Software Company were consulted simultaneously so as to clearly draw a line of distinction by the way these NFRs are logically dependent on each other. In creating SSIM [Figure 2(a)], four symbols (V, A, X, O) were used to represent the degree of coherence between two NFRs, *i* and *j* on the basis of set of rules mentioned below (Digalwar and Ganneri, 2015):

- 1 V – NFR *i* will help to achieve NFR *j*.
- 2 A – NFR *j* will help to achieve NFR *i*.
- 3 X – NFR *i* and *j* will help to achieve each other
- 4 O – NFR *i* and *j* are unrelated.

Reachability matrix can be then framed out from the SSIM and which can be then later analysed for transitivity.

### 3.3 Check transitivity between NFRs

The SSIM has been changed over into a binary grid, named initial reachability matrix as shown in Figure 2(b), by substituting V, A, X, O with either 1 or 0. By applying the tenets examined, an underlying reachability framework for the NFRs to execute COS is acquired. The final reachability Figure 2(c) matrix is then acquired by including transitivity.

Figure 2 Step by step illustration of ISM and MICMAC analysis

	N12	N11	N10	N9	N8	N7	N6	N5	N4	N3	N2	N1
N1	X	V	0	X	0	V	V	V	V	V	V	X
N2	V	V	0	X	0	V	V	V	V	V	V	X
N3	V	X	0	V	0	V	V	V	0	X		
N4	V	X	0	V	0	V	V	V	X			
N5	V	X	0	0	0	X	0	X				
N6	V	0	0	A	0	X	X					
N7	V	A	X	X	X	X						
N8	0	0	0	0	X							
N9	V	A	A	X								
N10	V	0	X									
N11	V	X										
N12	X											

(a)

	N12	N11	N10	N9	N8	N7	N6	N5	N4	N3	N2	N1
N1	X	V	0	X	0	V	V	V	V	V	V	X
N2	V	V	0	X	0	V	V	V	V	V	V	X
N3	V	X	0	V	0	V	V	V	0	X		
N4	V	X	0	V	0	V	V	V	X			
N5	V	X	0	0	0	X	0	X				
N6	V	0	0	A	0	X	X					
N7	V	A	X	X	X	X						
N8	0	0	0	0	X							
N9	V	A	A	X								
N10	V	0	X									
N11	V	X										
N12	X											

(b)

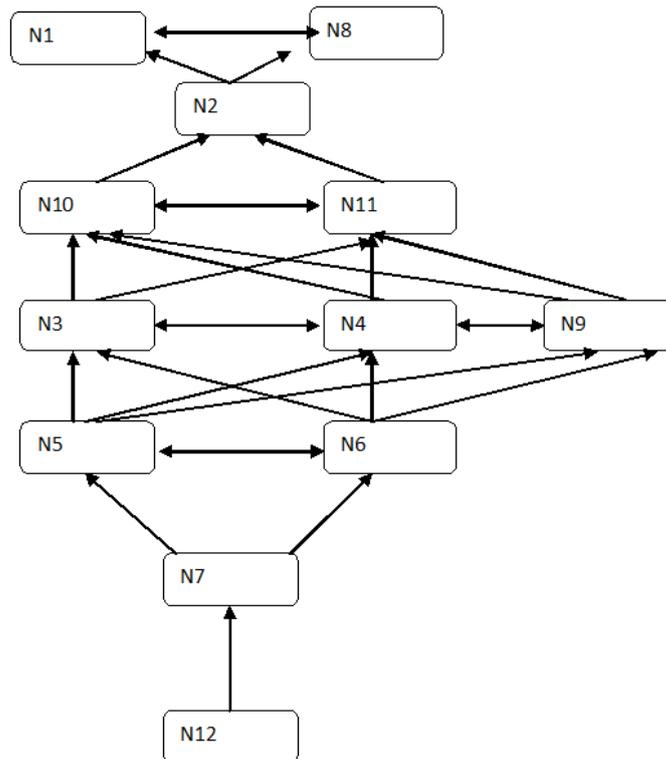
	N1	N2	N3	N4	N5	N6	N7	N8	N9	N10	N11	N12	Driver Power
N1	1	1	1	1	1	1	1	0	1	0	1	1	10
N2	0	1	1	1	1	1	1	0	1	0	1	1	9
N3	0	0	1	1	1	1	1	0	1	0	1	1	8
N4	0	0	1	1	1	1	1	0	1	0	1	1	8
N5	0	0	0	0	1	0	(1)	0	0	0	1	1	4
N6	0	0	0	0	0	1	1	0	(1)	0	0	1	4
N7	0	0	0	0	0	0	1	1	0	0	0	0	5
N8	0	0	0	0	0	0	1	1	0	0	0	0	2
N9	1	1	0	0	0	1	1	0	1	0	0	1	6
N10	0	0	0	0	0	0	1	0	1	0	0	1	5
N11	0	0	1	1	1	0	1	0	1	0	1	1	8
N12	1	0	0	0	0	0	0	0	0	0	0	1	2
Dependence Power	3	3	5	5	6	9	11	1	9	2	6	11	

(c)

**Figure 2** Step by step illustration of ISM and MICMAC analysis (continued)

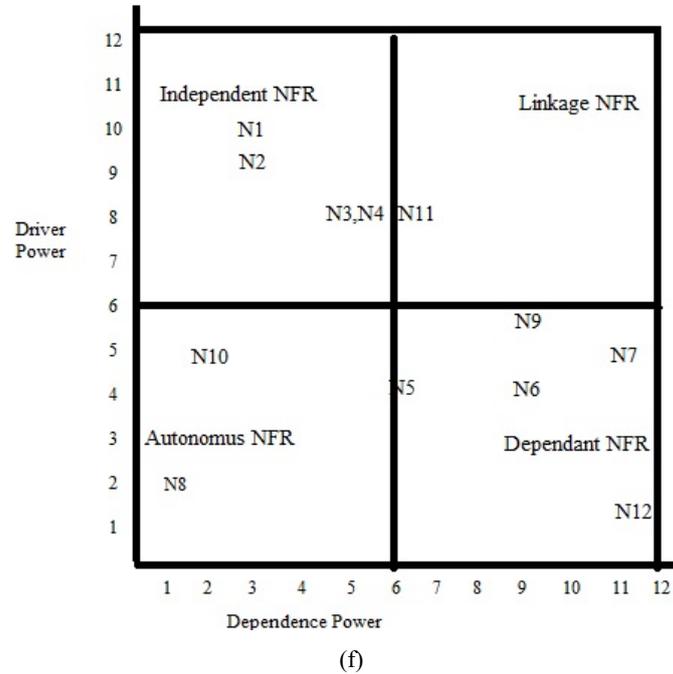
NFRs	Reachability set	Antecedent Set	Intersection Set	Level
N1	1,2,3,4,5,6,7,9,11,12	1,9,12	1,9,12	1
N2	2,3,4,5,6,7,9,11,12	1,2,9,12	2,9	2
N3	3,4,5,6,7,9,11,12	1,2,3,4,11	3,4,11	4
N4	3,4,5,6,7,9,11,12	1,2,3,4,11	3,4,11	4
N5	5,7,11,12	1,2,3,4,5,11	5,11	5
N6	6,7,9,12	1,2,3,4,6,7,9	6,7,9	5
N7	6,7,9,12	1,2,3,4,5,6,7,8,9,10,11	6,7,9	6
N8	7,8	8	8	1
N9	1,2,6,7,9,12	1,2,3,4,6,7,9,10	1,2,6,7,9	4
N10	7,9,10,12	7,10	7	3
N11	3,4,5,7,9,11,12	1,2,3,4,5,11	3,4,5,11	3
N12	1,12	1,2,3,4,5,6,7,9,10,11,12	1,12	7

(d)



(e)

**Figure 2** Step by step illustration of ISM and MICMAC analysis (continued)



### 3.4 Find level of each NFR and MICMAC analysis

Once the final reachability matrix has been designed, the reachability and predecessor set for each NFR were determined, which essentially incorporates NFR itself and other NFR's on which it might rely upon. Also on similar lines the antecedent set comprises of NFR itself and alternate NFR's which rely upon it. At this point, the convergence of these sets is inferred for all NFRs. The NFR for which the reachability and intersection point sets are same, assume top-level NFR position in the ISM hierarchy. Once the top level NFRs are recognised, they are then isolated from all other NFRs and, the same procedure is then rehashed for next level position. On the basis of the antecedent set and intersection set soft goal interdependency diagram is plotted in Figure 2(e). MICMAC graph [Figure 2(d)] is inferred by summing up the samples of all possible conceivable outcomes of interactions in the row termed as driving power and by computing the sum the sections of potential outcomes of collaborations columns terms as dependence power complied in the Figure 2(d). The terminology use for classification of NFRs is shown in Table 2. From the investigation when we map these NFRs, robustness, availability and security were observed to be the critical NFRs which cannot be overlooked regardless of situation which can easily compromised with Performance parameter which was found to be in category be less critical NFRs.

**Table 2** Different classification of NFRs made during MICMAC analysis

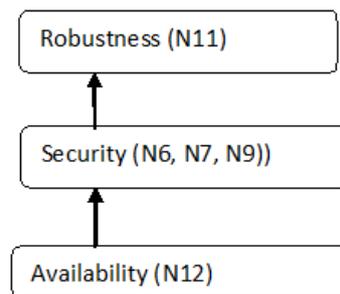
<i>Groups</i>	<i>Description</i>
Autonomous NFR (not critical)	The NFRs are relatively disconnected from the system framework thus can be ignored
Dependant NFRs (critical NFR)	These NFRs are totally dependant on other NFRs
Linkage NFRs (most critical NFRs)	Any activity on these NFRs will affect alternate NFRs furthermore a feedback impact on themselves, which may intensify with any moves or measures
Independent NFRs (less critical)	These NFRs can also be put on hold in case of limited resources.

*Source:* Booch et al. (1999)

### 3.5 Quantifying the link at each level of digraph

AHP's here is utilised to evaluate the priority of NFRs at every level in the resultant digraph (Saaty, 2008). Pairwise correlation between NFRs is performed at every level chosen by ISM. A pair wise comparison matrix **A** is then created using AHP, where **A** represents a real matrix of dimension  $m \times m$ , where  $m$  represents total evaluation criterions weighed. Every element  $a_{jk}$  of matrix **A**, speaks about the significance of  $j^{\text{th}}$  criterion in respect to  $k^{\text{th}}$  one subjected to condition that if  $a_{jk} < 1$ , then the  $j^{\text{th}}$  basis is less critical than the  $k^{\text{th}}$  paradigm. On the off chance that two criteria have the same significance, then the passage  $a_{jk}$  is 1, i.e.,  $a_{jk} \cdot a_{kj} = 1$  Consistency ratio has not been calculated as the consistency and transitivity has already been sorted by ISM.

The disentangled digraph after filtering out Autonomous NFRs and after removal of independent NFRs is shown in Figure 3. Final Simplified graph pin points that there is only one NFR at each level, but in case, if there are more than one NFR at each level then AHP may be applied to prioritise NFR at each of the level of hierarchy.

**Figure 3** Normalised softgoal interdependency digraph

### 3.6 Results

It is fascinating to contrast our assessment model and existing different strategies like the one proposed by Karlsson et al. (1998), where we found AHP system to be the most encouraging strategy to exchange off necessities that assisted exploration of its applicability however in our proposed approach, NFR's alongside their interdependencies were handled by applying coordinated methodology of ISM and AHP. On similar lines

(Kassab and Kilicay-Ergin, 2015) connected AHP to the goal model by mapping each OR-decomposition in the objective model into a distinct decision problem. They regard all non useful necessities as obligatory which is not essentially conceivable thus in our methodology we did MICMAC examination to discover basic NFRs so they are not disregarded even if there occurs a case of restricted assets. Zhu et al. (2012) proposes a fuzzy quantitative and quantitative soft goal interdependency charts (FQSIG) model with an objective of determining NFRs correlation analysis using trustworthy software and present the them using MATLAB-based tool. It utilises the relation matrix calculation that has extensively adapted to the negative and the positive effect on NFRs. This was further simplified by adopting MICMAC investigation and analysis methodology by out figuring reliance and driving force between NFRs.

**Table 3** Analysis of various ranking methods for NFRs based on multiple parameters

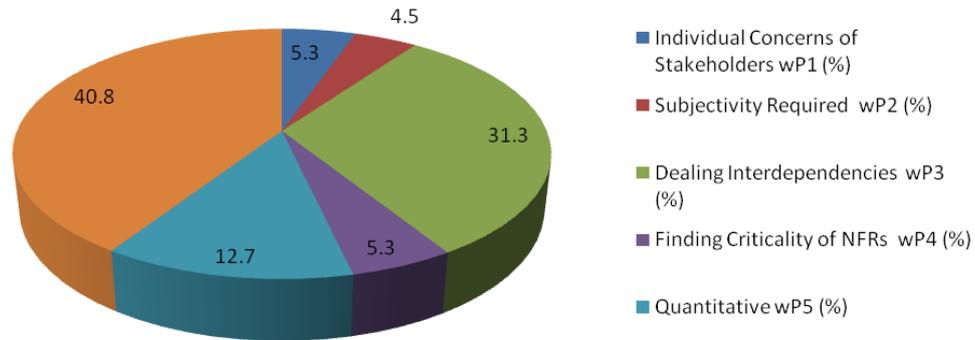
	<i>AHP</i>	<i>Fuzzy</i>	<i>Traceability matrix</i>	<i>Proposed approach</i>
Concept	Expressive	Expressive, logic-based, formal	Simple goal and rule-based	Visible, well-defined models
Individual concerns of stakeholders	Negligible	Negligible	Less	Negligible
Subjectivity	Very high	Very high	Less	Moderate
Dealing interdependencies	Yes	Limited	No	Yes
Find criticality of NFRs	Negligible	Negligible	Find critical NFRs	Find further categorisation of critical NFRs
Quantitative	Yes	Yes	No	Yes
Modelling structure to deal with interdependencies	Hierarchical tree	Does not deal	Does not deal	Interdependency diagraph

In this paper PriEst (Siraj, 2013), was used as an intuitive choice bolster tool to estimate needs from pairwise correlation judgments to contrast our approach with existing exchange off methodologies available in literature. as shown in Table 3. Table 4 consists of analysis of positioning techniques AHP, fuzzy, traceability matrix (TM) and proposed approach on the premise of six parameters.

**Table 4** Estimated values for the parameter weights

	<i>Individual concerns of stakeholders</i> $w_{P1}$ (%)	<i>Subjectivity required</i> $w_{P2}$ (%)	<i>Dealing interdependencies</i> $w_{P3}$ (%)
EV	5.3	4.5	31.3
GM	5.3	4.5	31.3
	<i>Finding criticality of NFRs</i> $w_{P4}$ (%)	<i>Quantitative</i> $w_{P5}$ (%)	<i>Modelling structure to deal with interdependencies</i> $w_{P6}$ (%)
EV	5.3	12.7	40.8
GM	5.3	12.7	40.8

**Figure 4** Chart on the basis of weights calculated for different parameters (see online version for colours)



In the first place the parameters are positioned with the assistance of PriEsT tool. The weights for parameters ascertained by utilising eigenvector (EV) technique and geometric mean (GM) are listed in Table 4. Managing Interdependencies (wP3) and modelling structure to manage interdependencies (wP6) were observed to be the most essential measure for looking at four different alternative methods for trade-off for NFR as appeared in Figure 4.

Proposed technique is considered the most preferred ranking technique on the basis of parameters (P1–P6) with weight of 46.7% followed by fuzzy with weight around 25.8% as shown in Table 5 and Table 6.

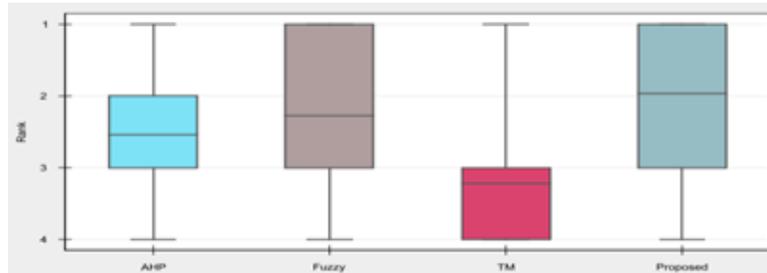
**Table 5** Estimated weights for the available ranking techniques for NFR

		<i>AHP</i>	<i>Fuzzy</i>	<i>Traceability matrix</i>	<i>Proposed technique</i>
P1	GM	0.25	0.25	0.25	0.25
	EV				
P2	GM	0.25	0.25	0.25	0.25
	EV				
P3	GM	.117	.223	.082	.578
	EV	.116	.223	.082	.578
P4	GM	.116	.122	.245	.517
	EV	.116	.122	.244	.518
P5	GM	.321	.321	.036	.321
	EV				
P6	GM	.17	.285	.073	.472
	EV	.17	.284	.073	.473
Overall		.177	.258	.098	.467

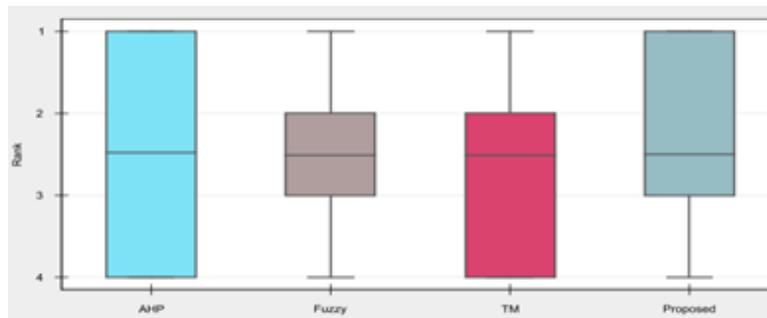
**Table 6** Weights suggested by PriEsT for selecting best trade-off technique

<i>Ranking techniques</i>	<i>AHP</i>	<i>Fuzzy</i>	<i>Traceability</i>	<i>Proposed</i>
%	17.7%	25.8%	9.8%	46.7

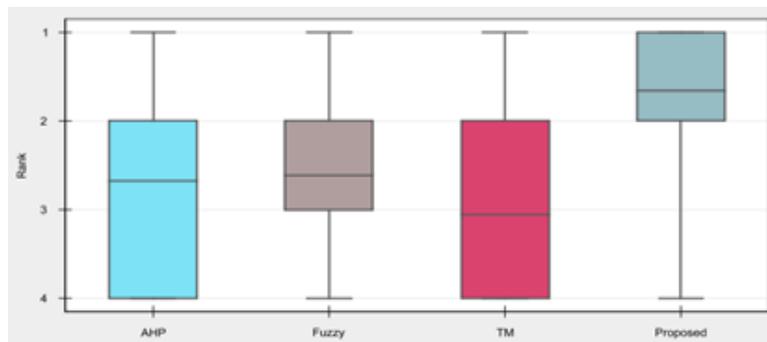
**Figure 5** Sensitivity analysis with uniform distribution by considering different parameters, (a) comparison on the basis of parameters P1, P2, P3 and P4 (b) comparison on the basis of parameters P3, P4, P5 and P6 (c) comparison on the basis of P3 and P6 (see online version for colours)



(a)



(b)



(c)

The methodology utilised for finding reasonable procedure is subjective in nature as there can possible irregularities in the judgments of the specialists. Such irregularities can be effectively tackled by applying multi-rule technique PrInt on PriEsT tool. Sensitivity investigation is accomplished by taking diverse arrangement of parameters at given time, as shown in Figure 5. The uniform distribution view in Figure 5(a) demonstrates fuzzy and proposed approach at the same level when thought about on the basis of parameters P1, P2, P3 and P4. Figure 5(b) indicates AHP and Proposed positioning equivalent if contrasted in terms of P3, P4, P5 and P6. Parameter P3 and P6 were weighted as vital

parameters in Table 5. At the same point when sensitivity investigation is done on the premise of these parameters, the proposed methodology is considered as the most favoured positioning strategy as shown in Figure 5(c). From above examinations, it's been reasoned that proposed approach to have list of critical NFRs has superseded the existed approaches available in literature on array of parameters and can be utilised to discover critical NFRs and model their interdependency on each other.

#### 4 Novel approach to specify identified critical NFRs in SRS

The specification of critical NFRs is a difficult task for the project team, because NFRs are subjective in nature. Modelling NFR with use case has been a focus point of numerous studies in the past as discussed in section 2 but because of great diversity in number and types of NFR it makes use case distracting and hard to read. As critical NFRs have been identified in Section 3, following terminology is used in this paper to model required NFRs.

- 1 **Kernel:** kernel deals with functional requirements of the system. Kernel capture system functionality from the perspective of actors. The key idea is to keep kernel simple and small by including only use cases with direct link to users. Mapping of NFR to existing use cases in the kernel is represented by adding another boundary (separate for each use case) outside the kernel. Each boundary deals separately with all the NFRs associated with that particular use case.
- 2 **Applicable condition:** applicable condition is associated with NFR which include constrain or condition required to implements NFR in system.
- 3 **NFRCase:** when a use case is identified having some associated NFR along with its applicable condition, it is must to define NFRCase that is required to control applicable condition. This is in order to mitigate or reduce the software risk. NFRCase triggers the action when Applicable condition fails.

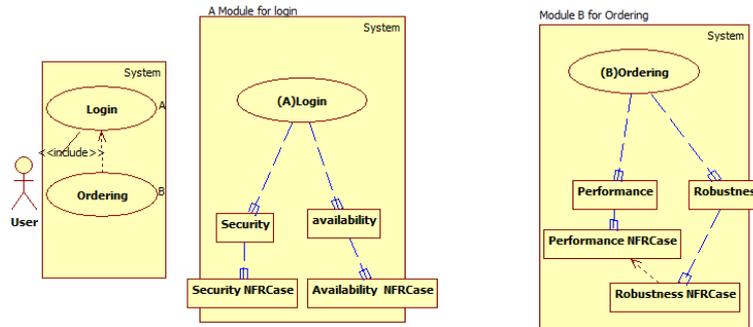
##### 4.1 Result analysis

In order to understand the new artefacts we have considered the same case of a simplified version of COS (Wiegiers and Beatty, 2013) that will let authorise employees to order meals from the company cafeteria online.

In Figure 6, UseCase has been designed on the basis of new artefacts, i.e., by making the Kernel simple and small. The two usecases login and ordering is shown with in the kernel. Both the use cases are linked to their extension with the name A and B module. In the extension module NFRs related to usecases and NFRCases are defined. NFR of usecases and NFRCase are different from each other. As shown in Figure 6 module A (extension to login use case) is linked to two NFRs security and availability (as defined in SRS). Further these NFRs have two modules. One is applicable condition (constraints or conditions required to implement NFR) which is defined outside the module boundary. Other is NFRCase which include the action need to take by system in case applicable condition is not satisfied. For the module A security NFRCase can be immediately logout the user if the connection seems insecure (may be because of identity of user or network

insecurity). Similarly availability NFRCase can be to display the page with text ‘service not available’.

**Figure 6** COS UseCase on the basis of proposed approach (see online version for colours)



Applicable Conditions For  
 Availability NFRCase: 5:00am and midnight(99.9%) & midnight and 5:00am(95%)  
 Security NFRCase:login is must for all operations&& only authorized users can access it.  
 Performance NFRCase:System shall display confirmation messages to users within 4 seconds after the user submits information to the system.  
 Robustness NFRCase:System shall enable the user to recover an incomplete order if system is broken prior to an order being either confirmed or canceled.

In order to analyse the validity of the proposed add-ons in SRS document, five SRS’s of various problem statements have been considered. The use case complexity and improved requirement-based complexity measure has been calculated for all the five SRS as shown in Table 7 and Table 8. Use case complexity (Booch et al., 1999; Kaur and Sharma, 2016) is calculated by:  $1 - \frac{\text{No. of UseCases} + \text{No. of NFRCases}}{\text{No. of Usage Relations} + \text{No. of Actors} + \text{No. of NFRs}}$  for

calculating the complexity on the basis of dependencies between use cases. Improved requirement-based complexity measure (Sharma and Kushwaha, 2010) which is defined as:  $IRBC = ((PC * PCA) + DCI + IFC)$  is derived from customer requirement on the basis of SRS document. The Computation of the IRBC for one of the SRS (Wieggers and Beatty, 2013), COS is shown below:

---

Inputs: 09
Outputs: 07
Number of interfaces: 07
Number of files: 01; IOC = 24
Number of functional req.: 39, FR = 1 * 39
Number of sub-processes 06;
$No. of NFR = 12, NFR = 3 * 1 + 4 * 2 + 5 * 3 = 26$
Requirement complexity = FR + NFR = 65
Product complexity, PC = (IOC * RC) = 24 * 65 = 1,560
Personal capability attrib. = 1; capability = nominal
No. of constraints 04
DCI = 1; IFC = 2
SFC = 2; no. of user and location, ULC = 6 * 2 = 12
Now, $IRBC = ((PC * PCA) + DCI + IFC;$

---

Complexity depends on how the NFRs are treated at the analysis phase. For example, IRBC for COS is 1,566 when focus is only on the critical NFR that is evaluated by ISM. These NFRs are must for the success of the proposed software. On the other hand complexity IRBC is 1,806 when we considered all the NFRs listed in SRS which are not important for the system. Many of NFRs are optional and can be put on hold. It is also not possible practically to deal all the NFRs at the same level.

**Table 7** Complexity without proposed add-ons

<i>SRS</i>	<i>Total NFRs</i>	<i>Use case complexity</i>	<i>IRBC</i>
Air traffic system	8	0.41	1,612
Meeting scheduler	6	0.5	1,014
Cafeteria ordering system	12	0.7	1,806
Telecommunication system	9	0.87	1,920
Verification of mobile phones	8	0.44	1,310

**Table 8** Complexity with proposed add-ons in SRS

<i>SRS</i>	<i>Total NFRs</i>	<i>Use case complexity</i>	<i>IRBC</i>
Air traffic system	3	0.25	1,418
Meeting scheduler	2	0.2	800
Cafeteria ordering system	5	0.14	1,566
Telecommunication system	4	0.35	1,704
Verification of mobile phones	4	0.29	1,217

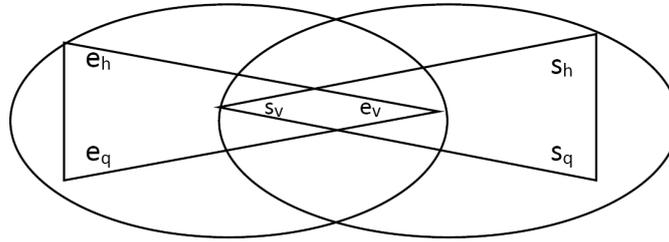
While calculating IRBC the measured complexity values increases when all 12 NFRs of COS are considered as important (which is practically impossible). During IRBC calculation NFR requirements are categorised into THREE categories with their associated precedence values as: optional = 1, important = 2, must = 3 which would further depend on the count. But complexity improves when NFRs are categorised by ISM technique. Hence, the incorporates proposed helps to reduce the complexity of NFRs measure as now focus is on critical NFRs. Similarly, when use case complexity is calculated on adding the aspect oriented use case model to the SRS it also improved as shown in Table 1 and Table 2. It has also been seen that proposed add-ons provides a more complete representation of the overall system without increasing the complexity from user point of view.

Artefacts and critical NFRs findings make it much more amenable to further simplification and integration as it is layered structure where as other approaches are not layered and difficult to integrate with other modelling techniques. But it has been seen from the past failures of software that an informal specification does not contribute much while validating the NFRs. There is need to have a formal model for validation of NFR but all the formal models in literature so far has validated only functional requirements as discussed in Section 2. The validation of NFR is done by using formal reference model as discussed in next section.

#### 4.2 Use of reference model for validation of NFR

Reference model has the application domain [environment (e)], the programming platform with its software [system (s)], and the interface between them. The quality artefacts are proposed to add into both environment ( $e_q$ ) and system ( $s_q$ ) division as shown in Figure 7 such that  $e = e_h \cup e_v \cup e_q$ ,  $s = s_h \cup s_v \cup s_q$ ,  $e_h \cap e_v \cap e_q = \Phi$ ,  $s_h \cap s_v \cap s_q = \Phi$ .  $e_q$  phenomenon is the part of environment which contributes to specify the quality of system. Similarly  $s_q$  is the part of system which contributes to specify quality of system.

**Figure 7** Reference model for NFR



Terminology  $e_h$ ,  $e_v$ ,  $e_q$ ,  $s_v$  are observable to user or environment where as  $s_h$ ,  $s_v$ ,  $s_q$ ,  $e_v$  are evident to system only. Therefore  $e_v$ ,  $s_v$  are observable to both environment and system. The designated terminology for COS (Wiegiers and Beatty, 2013) falls in six groups.

- $e_h$ : employee who order food, staff cooks and deliver food item.
- $e_v$ : ordering, making payment and confirmation.
- $s_h$ : internal representation of data flow.
- $s_v$ : beep at cafeteria.
- $e_q$ : speed of processing the order.
- $s_q$ : Response to orders. Gap between order and delivery time.

We validated the specifications of critical NFRS which with the help of environment properties, system properties and specifications:

- 1 Requirement allows that there should be consistency of W, P and M together. For availability  $\exists e_h e_q e_v s_v s_q$ . W should be consistent. To make the system available the desired property is  $\exists e_v. (\exists e_h e_q s. W) \Rightarrow (\exists e_h e_q s. W \wedge M \wedge P)$  which include strict response of environment to system behaviour.
- 2 A system is robust only if it can cope with errors during execution and cope with erroneous input, i.e.:
  - a  $\exists e. (\exists e_h e_q s. W) \Rightarrow (\exists s. S) \wedge (\forall s. S \Rightarrow (\exists e_h. W))$ .
  - b  $\exists e. (\exists s_h, s_v, s_q, e_v. S) \Rightarrow (\exists s. M \wedge P) \wedge (\forall s. (M \wedge P) \Rightarrow S)$ .
- 3 Performance of system is based on response time as:  $\forall e s. e_h e_q(t) - s_h(t) s_q(t) < t_{max}$  where value of  $t_{max}$  can be set on the basis of application. It will put restriction on the response time.

4 Security: if any user tried to access  $s_h$  files it will issue error.

Hence, considerable advantage can be gained from the adoption of formal model and techniques for NFRs if incorporate in SRS along with use case. It promotes correctness in development.

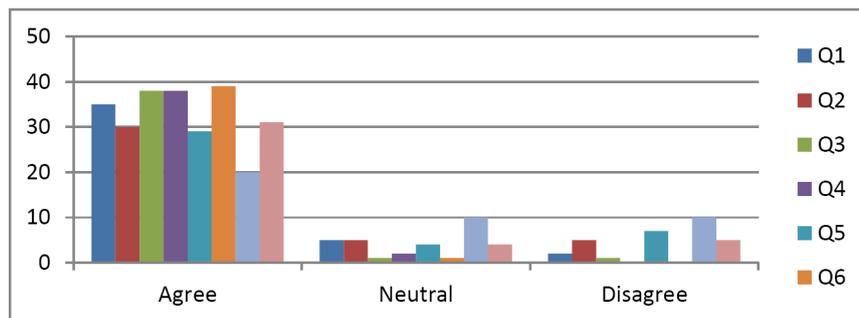
## 5 Evaluating usability of proposed artefacts in SRS

Further to understand the behaviour of our proposed add-ons, following set of questions is used for validating NFRs detail with the proposed incorporates as listed below:

- Q1 Does each requirement have a cause?
- Q2 Is each requirement practicable?
- Q3 Each requirement is testable once when put into execution?
- Q4 Is each requirement restricted and definite?
- Q5 Do any requirements have interdependency with other requirements?
- Q6 Is the requirement observable to goals of the system?
- Q7 Is the requirement is confined in quantitative terms?
- Q8 Are requirements stated undoubtedly? Can they be misapprehend?

Questions Q1–Q8 was taken from the Pressman (2005). We got 40 responses of stakeholders as shown in Figure 8 based on three level grading scales. The results indicate that the stakeholders find the incorporates suggested indeed resemble the desired NFRs. But, there were some stakeholders who feel that these were incompetent so some of them were unclear about their effectiveness.

**Figure 8** The results to questions Q1–Q8 (see online version for colours)



For the further insight into the distinguished features of the extended version of SRS following cognitive dimensions are considered as described below:

- Viable: is it able to implement critical NFRs by tracing back to previous phases?
- Traceable: is it easy to ascertain the successive stages in the development or progress?

- Bounded: measure for NFRs is defined?
- Visibility: is it easy to test the performance of NFRs representing through informally and formally?
- Effort: does it entail firm effort to do?
- Testable: can test cases for NFR be written with the help of SRS?
- Scope: boundaries of NFR are specified or not?
- Decision making: does NFR specification defend the basis of requirement importance when there is conflicting requirement?

Table 9 consists of analysis of standard SRS format and extended SRS on the premise of eight cognitive dimensions. In the first place the parameters are positioned with the assistance of PriEsT tool. The weights for parameters ascertained by utilising EV technique and GM are listed in Table 10 which shows effort and decision making to be the most essential measure for comparing standard and extended format of SRS.

**Table 9** Weights suggested by PriEsT for selecting format on the basis of cognitive dimensions

<i>SRS format</i>	<i>Weight (%)</i>
Extended	.676
Standard	.324

**Table 10** Estimated values for the cognitive dimensions

<i>Cognitive dimensions</i>	<i>Effort</i>	<i>Decision making</i>	<i>Scope</i>	<i>Bounded</i>	<i>Testable</i>
Weight (%)	.264	.189	.13	.11	.085
	.256	.185	.127	.109	.091
	.274	.191	.13	.103	.089
<i>Cognitive dimensions</i>	<i>Visibility</i>	<i>Traceable</i>	<i>Viable</i>		
Weight (%)	.084	.073	.065		GM
	.089	.073	.07		EV
	.085	.064	.064		PrinT

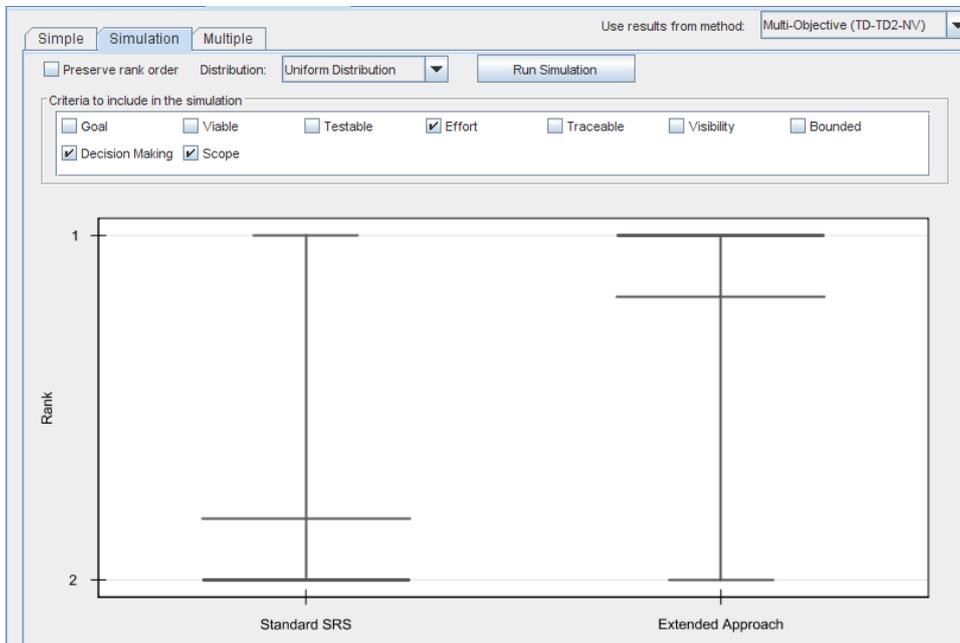
The methodology utilised for finding better format of SRS is subjective in nature as there can possible irregularities in the judgments of the specialists. Such irregularities can be effectively tackled by applying multi-rule technique PrInt on PriEsT tool as evaluated in Table 10.

Sensitivity investigation is accomplished by taking diverse arrangement of dimensions at given time. The Uniform Distribution view in Figure 9 demonstrates standard and extended format at the same level when thought about on the basis of all cognitive dimensions considered. But when both formats are compared on the basis of three cognitive dimensions effort, decision making and scope then extended measure proposed in this paper is favoured as shown in Figure 10. From above examinations, it's been reasoned that approach with new artefacts has supersedes the existed on array of dimensions and can be put into practice to deal with critical issues related to NFRs informal and formal modelling.

**Figure 9** Comparison on the basis of all cognitive dimensions considered (see online version for colours)



**Figure 10** Comparison on the basis of three cognitive dimensions effort, decision making and scope (see online version for colours)



## 6 Conclusions

The motivation for the work presented in this paper comes from the need to add details regarding NFRs for the proposed software's in SRS document. NFR concerns are ordinarily managed at configuration and usage level. Hence a model has been proposed that identifies critical NFRs from SRS by using ISM. Identified critical NFRs are then documented informally in SRS by use case which further is validated by formal reference model. The result obtained from the cognitive dimension analysis and complexity measures are able to establish that the proposed add-ons helps the software analysts to figure out all the concerns related to NFRs during the underlying periods of programming advancement.

## References

- Aburub, F., Odeh, M. and Beeson, I. (2007) 'Modelling non-functional requirements of business processes', *Information and Software Technology*, November, Vol. 49, Nos. 11–12, pp.1162–1171.
- Berenbach, B. and Gall, M. (2006) 'Toward a unified model for requirements engineering', *ICGSE*, pp.237–238.
- Boehm, B. et al. (1976) 'Quantitative evaluation of software quality', *Proceedings of the 2nd International Conference on Software Engineering*, IEEE, pp.592–605.
- Booch, G., Rumbaugh, J. and Jacobson, I. (1999) *The Unified Modeling Language User Guide*, Addison-Wesley, USA.
- Chung, L., Nixon, B., Yu, E. and Mylopoulos, J. (2009) 'Non-functional requirements in software engineering', in *Conceptual Modeling: Foundations and Applications*, pp.363–379, Springer, Heidelberg.
- Cysneiros, L.M. and Sampaio do Prado Leite, J.C. (2004) 'Nonfunctional requirements: from elicitation to conceptual models', *IEEE Transactions on Software Engineering*, May, Vol. 30, No. 5, pp.328–350 [online] <http://dx.doi.org/10.1109/TSE.2004.10>.
- Digalwar, A.K. and Ganneri, G. (2015) 'Interpretive structural modeling approach for development of electric vehicle market in India', *Procedia CIRP Elsevier*, Vol. 26, pp.40–45 [online] <https://doi.org/10.1016/j.procir.2014.07.125>.
- Elahi, M. and Yu, Y. (2011) 'A semi-automated tool for requirements trade-off analysis', *011 IEEE 35th Annual Computer Software and Applications Conference*, Munich, pp.466–475.
- Ernst, N.A., Yu, Y. and Mylopoulos, J. (2006) 'Visualizing non-functional requirements', *Requirements Engineering Visualization, REV'06, First International Workshop*, September, p.2.
- Fan, Z-P. and Liu, Y. (2010) 'A method for group decision-making based on multi-granularity uncertain linguistic information', *Expert Systems with Applications: An International Journal*, May, Vol. 37, No. 5, pp.4000–4008, ACM.
- Firesmith, D. (2004) 'Prioritizing requirements', *Journal of Object Technology*, September–October, Vol. 3, No. 8, pp.35–47.
- Fuentes-Fernández, L. and Vallecillo, A. (2004) 'An introduction to UML profiles', *The European Journal for the Informatics Professional*, Vol. 5, No. 2, pp.5–13.
- Glinz, M. (2007) 'On non-functional requirements', in *15th IEEE International Requirements Engineering Conference (RE'07)*, pp.21–26.
- Grady, R. and Caswell, D. (1987) *Software Metrics: Establishing a Company-wide Program*, Prentice-Hall, Englewood Cliffs.

- Gunter, C., Gunter, E., Jackson, M. and Zave, P. (2000) 'A reference model for requirements and specifications', *IEEE Software*, May–June, Vol. 17, No. 3, pp.37–43.
- IEEE (1993) Recommended Practice for Software Requirements Specifications, IEEE Standard 830-1993 (please check the format).
- ISO/IEC 25010:2011(E): Software Engineering – Product Quality – Part 1: Quality Model (2011).
- Jacobson, I. and Cook, S. (2010) *Blog on 'The Road Ahead UML' on Dobb's*, 12 May.
- Jureta, I.J., Faulkner, S. and Schobbens, P.-Y. (2006) 'A more expressive softgoal conceptualization for quality requirements analysis', *LNCSE Springer*, Vol. 4215, pp.281–295.
- Jürjens, J. (2002) 'UMLsec: extending UML for secure systems development in UML', *Proceedings of the 5th International Conference on The Unified Modeling Language*, London, UK, Springer-Verlag, pp.412–425.
- Karlsson et al. (1998) 'An evaluation of methods for prioritizing software requirements', *Information and Software Technology*, Vol. 39, Nos. 14–15, pp.939–947.
- Kassab, M. and Kilicay-Ergin, N. (2015) 'Applying analytical hierarchy process to system quality requirements prioritization', *Innovations in Systems and Software Engineering*, December, Vol. 11, No. 4, pp.303–312.
- Kaur, H. and Sharma, A. (2016) 'A measure for modelling non-functional requirements using extended use case', *Proceedings of the 10th INDIACom, IEEE INDIACom-2016, 3rd 2016 International Conference on 'Computing for Sustainable Global Development'*, 16–18 March.
- Liaskos, S. et al. (2013) 'On eliciting contribution measures in goal models', *7th International Conference on Research Challenges in Information Science (RCIS)*, IEEE, Paris, pp.1–8.
- Lodderstedt, T., Basin, D.A. and Dosser, J. (2002) 'SecureUML: a UML based modeling language for model-driven security', in *UML'02: Proceedings of the 5th International Conference on the Unified Modeling Language*, London, UK, Springer-Verlag, pp.426–441.
- Mairiza, D. et al. (2010) 'An investigation into the notion of non-functional requirements', *Proceedings of SAC'10*, ACM, pp.311–317.
- McCall, J.A. and Matsumoto, M.T. (1980) *Software Quality Measurement Manual*, Vol. II, Rome Air Development Center, RADC-TR-80-109-Vol-2.
- Miller, S.P. and Tribble, A.C. (2001) 'Extending the four-variable model to bridge the system-software gap', *Digital Avionics Systems, DASC, 20th Conference*, Daytona Beach, FL, IEEE, Vol. 1, pp.4E5/1–4E5/11.
- Moreira, A. et al. (2005) 'Multi-dimensional separation of concerns in requirements engineering', *13th IEEE International Conference on Requirements Engineering (RE'05)*, IEEE, pp.285–296.
- Mylopoulos, F. et al. (1992) 'Representing and using nonfunctional requirements: a process-oriented approach', *IEEE Transactions on Software Engineering*, June, Vol. 18, No. 6, pp.483–497.
- Pavlovski, C.J.J. and Zou (2008) 'Non-functional requirements in business process modeling', *Proc. of 5th Asia-Pacific Conference on Conceptual Modelling*, Wollongong, NSW, Australia, CRPIT, Vol. 79, pp.103–112.
- Pressman, R.S. (2005) *Software Engineering: A Practitioner's Approach*, 6th ed., McGraw-Hill Publication, New York.
- Rao, A.A. and Gopichand, M. (2011) 'Four layered approach to non-functional requirements analysis', *IJCSI International Journal of Computer Science Issues*, November, Vol. 8, No. 6, pp.371–379.
- Roman, G.-C. (1985) 'A taxonomology of current issues in requirements engineering', *IEEE Computer*, April, Vol. 18, No. 4, pp.14–21.
- Saaty, T.L. (2008) 'Relative measurement and its generalization in decision making: why pairwise comparisons are central in mathematics for the measurement of intangible factors – the analytic hierarchy/network process', *Real Academia de Ciencias*, Vol. 102, No. 2, pp.251–318, Espana.

- Sharma, A. and Kushwaha, D.S. (2010) 'A complexity measure based on requirement engineering document', *Journal of Computer Science Engineering*, Vol. 1, No. 1, pp.112–117.
- Siraj, S. (2013) 'PriEsT: an interactive decision support tool to estimate priorities from pairwise comparison judgments', *Intl. Trans. in Op. Res.*, pp.1–19.
- Sommerville, I. (2006) *Software Engineering*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA.
- Supakkul, S. and Chung, L. (2004) 'Integrating FRs and NFRs: a use case and goal driven approach', *Proc. SERA 04*, IEEE, pp.30–37.
- Tonu Subrina, A. and Tahvildari, L. (2005) 'Towards a framework to incorporate', in *Proceedings of IEEE WCRE Workshop on Reverse Engineering to Requirements (RETR)*, Pittsburgh, Pennsylvania, USA, November, pp.13–18.
- Warmer, J. and Objecten, K. (2015) *The Future of UML* [online] <http://www.klasse.nl/english/uml/uml2.pdf> (accessed November 2015).
- Wieggers, K. and Beatty J. (2013) *Software Requirements*, Pearson Education, Redmond, WA, USA.
- Yen, J. and Tiao, W.A. (1997) 'A systematic trade-off analysis for conflicting imprecise requirements', *3rd IEEE International Symposium on Requirements Engineering*, IEEE, pp.87–96.
- Zheng, X., Liu, X. and Liu, S. (2010) 'Use case and non-functional scenario template-based approach to identify aspects', *Computer Engineering and Applications (ICCEA), 2010 Second International Conference*, 19–21 March, Vol. 2, pp.89, 93.
- Zhu, M-X., Luo, X-X., Chen, X-H. and Wu, D.D. (2012) 'A non-functional requirements tradeoff model in trustworthy software', *Information Sciences: An International Journal*, May, Vol. 191, pp.61–75, Elsevier.