A Systematic Classification and Analysis of NFRs

Pericles Loucopoulos  
Harokopio University of Athens  
p.loucopoulos@hua.gr

Jie Sun  
Zhejiang University  
jianbin@zju.edu.cn

Liping Zhao  
The University of Manchester  
liping.zhao1@gmail.com

Farideh Heidari  
Delft University of Technology  
f.heidari@tudelft.nl

ABSTRACT

The main agenda of Requirements Engineering (RE) is the development of tools, techniques and languages for the elicitation, specification, negotiation, and validation of software requirements. However, this development has traditionally been focused on functional requirements (FRs), rather than non-functional requirements (NFRs). Consequently, NFR approaches developed over the years have been fragmental and there is a lack of clear understanding of the positions of these approaches in the RE process. This paper provides a systematic classification and analysis of 89 NFR approaches.

Keywords
Non-functional requirements; NFRs; NFR approaches; quality requirements; requirements engineering; systematic classification and analysis of NFRs approaches.

INTRODUCTION

The importance of Requirements Engineering (RE) has long been established both in research and in practice. There is a significant intellectual activity in the RE field as evidenced by the large volume of papers published in journals and conferences c.f. [1] and by the adoption of a variety of techniques by industry c.f. [2]. A number of initiatives have been put forward to support the RE lifecycle, initiatives such as risk-driven methodologies [3], requirements tracing [4, 5], model re-use [6], use of scenarios [7], use of visualization [8], use of business rules [9, 10], enterprise modeling [11] and goal modeling [12, 13] to name but a few.

Historically, there has been a distinction between two classes of requirement, functional requirements (FRs) and non-functional requirements (NFRs), the former referring to the operational properties and the latter to the quality properties that the desired system must possess. It is generally agreed that more emphasis has been placed to date on FRs rather than NFRs. However, the changing landscape of requirements demands an increased attention to those aspects of requirements that deal with quality factors, the NFRs. There is a whole set of new advanced design requirements [14] that demand now more than ever close attention to the effectiveness of the SE process. The field’s focus and scope have shifted from engineering of individual systems and components towards the generation, adaptation and maintenance of software-intensive ecosystems consisting of software, hardware, human and organizational agents, business processes and more. Such software ecosystems require attention in three areas namely those of design and evolution, orchestration and control, and monitoring and assessment [15].

The context and motivation for this paper is the changing landscape of RE that demands now more than ever particular attention to NFRs. Whilst there is ample evidence (c.f. [16]) that NFRs play a significant role in systems, there is surprisingly an absence of an agreed position regarding the definition of NFRs, their classification and presentation [17].

To address this problem, the paper presents a systematic literature review (SLR) of the field of NFRs. Recent years have witnessed an increase in the use of evidence based approaches in SE [18, 19] with a SLR c.f. [20] being regarded as the most reliable method. To this end procedures for evidence-based SE are presented in [21, 22]. A study of 33 unique studies, carried out between 2004-2007, revealed 35 SLRs in the field of SE [23]. This paper provides an aggregated view about research efforts to date on NFRs.

The paper is organized as follows. First, the methodology adopted for the classification framework is defined, introducing also the classification scheme. Second, using this classification scheme the 89 approaches are defined according to their relevance in each one of the possible classes of evaluation. Third, an analysis is provided in terms of the contribution of the
89 approaches. Finally, the paper concludes with some reflections and a number of observations for future research challenges in the field of NFRs.

**METHODOLOGY FOR THE CLASSIFICATION SCHEME**

Kitchenham et al [23] identify two types of SLR, conventional SLRs, relating to a specific research question and mapping SLRs, concerned with identifying and classifying primary studies in an area. This paper falls in the mapping category. Publications related to NFRs were identified by (a) searching databases and citation indices (DBLP, Google Scholar), (b) searching journals (most prominently, Requirements Engineering journal, IEEE Software, IEEE Transactions on Software Engineering) and (c) searching the proceedings of relevant conferences (most prominently, IEEE International RE Conference, International Conference on SE, International Conference on Conceptual Modeling, AMCIS, ICIS). This search revealed 89 candidate approaches.

In order to analyze these approaches a classification scheme was devised that is based on three fundamental perspectives within RE [24]: (a) the contextual dimension; (b) the process dimension; and (c) the application dimension. The choice of these three dimensions as a framework for reviewing NFRS is grounded on the universally accepted methodology of RE in terms of its activities, augmented by two considerations that consider the focus and application of NFRs.

The contextual dimension addresses a dichotomy that characterizes requirements as those of early and late requirements. Early requirements [25-27] are almost exclusively driven by client stakeholders’ communication. Issues of early requirements include: (a) the customer profiles of a business process, (b) the likely demand for product or service made by each type of customer, (c) the level of desirable service that the business process should strive to achieve, (d) the resources that are required in order to achieve these levels of service and (e) the trade-off between levels of service and requisite resource between all client stakeholders of a business process. Late requirements focus on the attributes of the desired system in terms of both functionality and quality.

The process dimension addresses the issue of where a particular approach fits within the RE lifecycle. Within the RE literature, multiple frameworks have been developed, positing anywhere from two to seven primary requirements tasks [28]. For the current review, we adopt a widely-employed and straightforward categorization of the requirements processes into four facets: (i) discovery (more commonly referred to as elicitation), (ii) specification, (iii) negotiation and (iv) validation & verification [29]. During requirements discovery, designers develop an understanding of the application domain and infer specific design needs through consultation with stakeholders and other sources [30]. Requirements specification is both a noun and a verb in RE parlance. As a noun, a specification is the document in which the requirements for a design effort are articulated thus representing an agreement between the stakeholders and the design team [31, 32]. The verb suggests the process of developing and managing the specification document; it is the process by which the design team acquires, abstracts, and represents the requirements [29, 33]. During requirements negotiation stakeholders focus on alternative solutions and attempt to envision potential behavior of the system prior to its implementation [34]. Finally, during requirements validation and verification designers ensure that the requirements are (i) of high quality, (ii) by addressing the users’ needs, (iii) being appropriate for the design effort, and (iv) having no inconsistencies or defects [35].

The application dimension is used to classify approaches to those that focus on either one or more specific NFRs (e.g. security, privacy and performance) or on approaches that are of generic nature that may be used for any kind of NFR.

The 89 approaches reported in this paper were investigated against these three dimensions which when considered in their totality give rise to 16 classification criteria.

**CLASSIFICATION OF NFR APPROACHES**

This section provides a classification of known NFR approaches, using the multi-dimensional framework described in the preceding section. The results are shown in tabular format in this section. For reasons of readability the overall table is presented in four parts according to the four phases of the process dimension. For each process phase the investigated approaches are classified according to whether they address early or late requirements and for each one of there is a further classification based on the application whether it is aimed at a specific quality characteristic or the approach is generic and therefore potentially applicable to any application. Each entry is given a name and at least one representative reference. Some entries are named according to the name given by the authors of the approach whereas others are introduced in this paper so that overall there is consistency and also an easy way of distinguishing one from another beyond the actual reference.

Table 1 gives a summary of the investigated approaches for the elicitation of NFRs. There are altogether 21 approaches with greater emphasis on generic applicability.

Most approaches that deal with early requirements (i.e. examining NFRs from an organizational prism) are goal-oriented, and
they usually treat NFRs as soft-goals.

For example, the i* framework [36], founded on the concepts of goal and agent, is capable of capturing organizational knowledge of soft-goals through the identification of organizational agents and their dependencies. The NFR Framework [37-39] elicits soft-goals through a Soft-goal Interdependency Graph. The EKD approach [40] proposes the enterprise goal view, and based on this view, uses a network of goals to capture the causal structure of an enterprise.

<table>
<thead>
<tr>
<th>Early Requirements</th>
<th>Generic</th>
<th>Late Requirements</th>
<th>Specific</th>
<th>Generic</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Security Misuse Cases [41], [42]</td>
<td>• i* Framework [36]</td>
<td>• Use Cases for Efficiency NFR [49].</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Usability Soft goal Catalogue [43].</td>
<td>• NFR Framework [37], [38], [39]</td>
<td>• NFR Framework [37], [38], [39]</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• EKD [40]</td>
<td>• Quality-Based RE Framework [44]</td>
<td>• Quality-Based RE Framework [44]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Quality-Based RE Framework [44]</td>
<td>• Activity-Based Quality Model [45]</td>
<td>• Activity-Based Quality Model [45]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Activity-Based Quality Model [45]</td>
<td>• NFR Enhanced Problem Frames [46]</td>
<td>• NFR Enhanced Problem Frames [46]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Problem Frames for Soft goals [47]</td>
<td>• Layered NFRs Analysis [48].</td>
<td>• Layered NFRs Analysis [48].</td>
<td></td>
</tr>
</tbody>
</table>

Table 1. Elicitation Process

For late requirements, there has been much research activity since historically this was the area that first attracted attention by researchers concerned with software quality. In general these approaches fall into three categories: (i) stakeholder engagement [50], [58], (ii) use of patterns [51, 52] and models [53-55], and (iii) identification of NFRs from textual descriptions [56, 57].

Table 2 gives a summary of the investigated approaches for the specification of NFRs. There are altogether 30 approaches with greater emphasis on late requirements.

Early requirements approaches for NFRs are mostly generic techniques the majority of which are based on some variation of the goal-oriented paradigm. The NFR Framework [37-39] and i* framework [36] are the most popular specification approaches for generic NFRs. The former uses Soft-goal Interdependency Graphs to specify decompositions and dependencies of NFRs; whereas the latter provides a strategic dependency model and a strategic rationale model that specify early NFRs graphically. Based on the soft-goal concept, there are also textual specification language, such as the informal requirements specification table used by Yin and Jin [46] to represent characteristics and conditions of NFRs, the semi-formal textual specification proposed by Jureta et al. [59] to characterize NFRs, and Formal Tropos [60] that complements graphical i* models with a formal notation.

With regards to late requirements, there have been a number of approaches that focus on specific applications, primarily those of (a) security c.f. the works on security use/misuse cases [61, 62], security activity diagrams [63], and security patterns [64]; (b) performance c.f. the work on UML-centric approaches [65], semantic frameworks [66], and agile software performance specification [67]; (c) efficiency, by integrating it into use cases c.f. [49].

A large number of works that specify generic NFRs have been proposed in the literature. Late NFRs are specified and represented based on UML models [68-73], goal models [51, 74-77], use cases [51, 77, 78], and process models [79-81].

There are also other languages and templates for NFR specification, such as those put forward by [82], quality attributes taxonomy [83] and the Volere Specification Template [84].

Table 3 gives a summary of the investigated approaches focusing on the negotiation of NFRs. There are altogether 16 approaches that deal with negotiation issues, the majority of which focus on late requirements with the exception of the use of matrix maps [78], goals [85, 86] and conflict resolution [87].

For the negotiation of specific NFRs, there have been several approaches. To support the identification and resolution between different NFRs, a set of approaches and tools have been proposed. These include the cost-quality tradeoff approach.
[88], the Quality Attribute Risk and Conflict Consultant tool [89] and the Software Cost Option Strategy Tool [90]. For trustworthy software, Zhu et al [91] developed a fuzzy qualitative and quantitative soft-goal interdependency graphs approach (FQSIG). To manage conflicts between security and usability, Mairiza and Zowghi [92] proposed an ontological framework.

<table>
<thead>
<tr>
<th>Early Requirements Specific</th>
<th>Late Requirements Specific</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Security Misuse Cases[41],[42]</td>
<td>• Security Use Cases [61]</td>
</tr>
<tr>
<td>• i* Framework [36]</td>
<td>• Security Activity Diagram [63]</td>
</tr>
<tr>
<td>• NFR Enhanced Problem Frames [46]</td>
<td>• Security Patterns [64]</td>
</tr>
<tr>
<td>• Quality Requirements Analysis [59]</td>
<td>• UML Based Performance Engineering [65]</td>
</tr>
<tr>
<td>• Formal Tropos [60]</td>
<td>• Semantic Framework for Performance [66]</td>
</tr>
<tr>
<td>• Agile Software Performance Spec [67]</td>
<td>• Use Cases for Efficiency NFR [49]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Early Requirements Specific</th>
<th>Late Requirements Specific</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Matrix Maps [85]</td>
<td>• Cost-Quality Tradeoff [88]</td>
</tr>
<tr>
<td>• GBRAM [86]</td>
<td>• Quality Requirements Conflict Identification [89]</td>
</tr>
<tr>
<td>• NFR Conflict Resolution [87]</td>
<td>• Software Cost Option Strategy Tool [90]</td>
</tr>
<tr>
<td>• FQSIG [91]</td>
<td>• Ontological Framework for Security &amp; Usability [92]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Early Requirements Specific</th>
<th>Late Requirements Specific</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Groupware for Requirements Negotiation [93],[94]</td>
<td>• A Catalog of Conflicts [95]</td>
</tr>
<tr>
<td>• Ontology-Based Conflict Analysis [96]</td>
<td>• Extended NFR Framework [76], [97]</td>
</tr>
<tr>
<td>• NFR Optimization [98]</td>
<td>• NFR Traceability [99],[100]</td>
</tr>
<tr>
<td>• Hierarchical Constraint Logic Programming [101]</td>
<td>• Personal Construct Theory [102]</td>
</tr>
</tbody>
</table>

Table 2. Specification Process

Table 3. Negotiation Process

To manage conflicts among late NFRs, there are a variety of generic approaches including conflict identification, analysis and resolution. The most prominent approach is the Win-Win groupware system [93, 94]. It determines both FRs and NFRs as negotiated win conditions, based on the WinWin Spiral Model. There are other approaches also such as those using conflict catalogues [95] and ontology [96]. The NFR Framework, already presented in the elicitation and specification tables has been widely extended [76, 97] and adapted [98] in different ways to identify, analyze and resolve various NFR conflicts. Modeling requirements traceability is another method to address NFR conflicts [99, 100]. Other theories include Hierarchical Constraint Logic Programming [101] and Personal Construct Theory [102].

Table 4 gives a summary of the investigated approaches focusing on the validation and verification of NFRs. There are altogether 22 approaches that address issues of validation and verification.

For early requirements there are approaches that deal with specific applications, specifically, with reliability [103] using UML models and with performance by adapting the NFR Framework [104], the later also being of relevance to late...
requirements. Within early requirements there are many approaches that are of generic nature of which the NFR Framework is used [37-39] to provide a label propagation mechanism to qualitatively evaluate NFR achievement. The label propagation mechanism has been extended to support quantitative evaluation [105-107]. The validation check list [48] and traceability management [108] are also frequently-used to validate NFRs. All these approaches, except [48], can be used for late NFRs as well.

<table>
<thead>
<tr>
<th>Early Requirements</th>
<th>Late Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Specific</strong></td>
<td><strong>Generic</strong></td>
</tr>
<tr>
<td>* Bayesian Reliability Prediction [103]</td>
<td>* NFR Framework [37],[38],[39]</td>
</tr>
<tr>
<td>* Performance Requirements Framework [104]</td>
<td>* Quantitative NFR Patterns [105]</td>
</tr>
<tr>
<td>* Quantitative NFR</td>
<td>* Quantitative Assessment of NFRs [106]</td>
</tr>
<tr>
<td>* Assessment of NFRs [106]</td>
<td>* Reasoning About Partial Goal Satisfaction [107]</td>
</tr>
<tr>
<td>* Partial Goal Satisfaction [107]</td>
<td>* Traceability Management [108]</td>
</tr>
<tr>
<td>* Validation Check List [48]</td>
<td>* UML-Based Verification [111]</td>
</tr>
<tr>
<td>* Traceability Management [108]</td>
<td>* Scenario-Based Assessment [112]</td>
</tr>
<tr>
<td></td>
<td>* Bayesian Belief Network [113]</td>
</tr>
<tr>
<td></td>
<td>* Contextual Information Ontology [114]</td>
</tr>
<tr>
<td></td>
<td>* Automata-Based Verification [115]</td>
</tr>
<tr>
<td></td>
<td>* Abstract Interpretation Based Verification [116]</td>
</tr>
</tbody>
</table>

Table 4 Validation and Verification Process

For late requirements, specific applications have been targeted such as performance [109] and reliability [110] of systems. There are also approaches that have been proposed from a general perspective dealing independently of any application with issues such as verification using UML models [111] or using scenarios for assessing the degree of validity of NFRs [112], or using formal methods, such as Bayesian belief network [113], ontology [114], finite state automata [115], and abstract interpretation [116] for verification purposes.

**ANALYSIS OF NFR APPROACHES BASED ON THE CLASSIFICATION SCHEME**

Using the above classification, three different analysis were performed: (a) process-oriented analysis to analyze the numeric impact of approaches in each one of the 16 different options, (b) single dimensional analysis to examine the percentage of approaches focused along each one of the three dimensions and (c) multi-aspect analysis within a single dimension to demonstrate the degree to which investigated approaches address than one aspect within a dimension.

Figure 1 demonstrates the process-oriented analysis. This presentation distinguishes a number of approaches in each process within different aspects of the other two dimensions namely, context and application. In other words, the number of approaches covering each combination of <process, application (specific/generic)> and <process, context (early/late)> are provided in figure 1. The analysis is in terms of absolute values.

This analysis reveals that, in all 4 phases generic approaches are much more prominent. This is not surprising since most new research focuses on the provision of theories, models and techniques at an abstract level. However, the paucity of approaches that tackle specific aspects of quality is a cause for concern because ultimately, NFRs must address specific quality requirements whether early or late, requirements such as compliance, security, privacy, reliability, robustness and so on.

The graph of Figure 1 also reveals that there is more emphasis on late requirements rather than early requirements, right across all 4 phases of the NFRs lifecycle. In other words, the emphasis is mostly on requirements for software and systems rather than on organizational and human factors. And yet research and practice have shown that there is an intertwining between requirements for systems and requirements for contexts of systems [14].

Figure 2 summarizes the single-dimension analysis and compliments the findings presented in Figure 1, by presenting a comparative analysis for context, process and application. For each dimension, the percentage of approaches focused on different aspects of the dimension is presented.
For context and application the comparative influences have already been discussed. For the process dimension, the specification phase is the phase that has received most attention and negotiation the least. The bias towards specification is perhaps due to the preference of researchers in the area to focus on the innovation of new modeling techniques rather than any on the other issues. However, the relatively small effort in negotiation, coupled with the lack of effort in early requirements seems to indicate that human-centered and contextual issues do not receive as much attention as they should.

Figure 3 summarizes the analysis of two dimensions namely process and context. The investigated approaches were analyzed to ascertain to what degree an approach covers one or more of the process phases. The analysis also focused on the degrees to which approaches focused on one or the other of the two contexts and also those that consider both contexts.

![Figure 1. Process-oriented analysis](image1)

![Figure 2. Analysis against three dimensions](image2)

The graphs of Figure 3 reveal the fragmentation of the field in both process and context. In terms of process, the majority of approaches (87%) deal with only one of the well-accepted phases of the requirements lifecycle. A small number of approaches (4%) deal with 3 of the 4 phases. In general, it seems that there is scope for the development of approaches that embrace all lifecycle phases. This is a methodological issue that needs some attention. In terms of context only a small number of approaches (7%) cover both early and late requirements in other words having the ability to seamlessly move from human-centered to system-centered NFRs.
CONCLUSIONS

The main contribution of this paper is the establishment of a clear understanding of the NFR approaches in RE. The paper has presented this understanding through a systematic classification and analysis of 89 NFR approaches. Two important conclusions can be drawn from this analysis:

- **Lack of NFR approaches for early requirements.** One important finding of this paper is that most current NFR approaches are for late rather than early requirements. This seems to suggest that early requirements are more challenging, as they tend to be informal and expressed in natural language. Consequently, they are difficult to systematize. Future research should look into text analysis and natural language processing approaches for the treatment of early NFRs.

- **Lack of domain-specific NFR approaches.** Among 89 NFR approaches we have sampled, only 18 are domain-specific approaches. It seems that the field has not settled yet as to whether generic approaches can address NFRs for all application domains or there is a need for specific NFR approaches for different categories of application (e.g. security, privacy, compliance etc.). Future research needs to focus on the ontology of different applications and through this to ascertain whether generic approaches are adequate for covering all quality aspects of such application domains or there is a need for focusing on specific approaches.

There are a number of implications from this analysis regarding the utility of NFRs with respect to technological advances in the field of information systems, challenges emerging from areas such as service computing, cloud computing and mobile computing.

Indeed, our initial finding [117] shows that most of today’s service system development approaches focus mainly on the system facing quality requirements (the *late NFRs*), rather than the customer facing quality requirements (the *early NFRs*). This suggests that despite of the latest advances in the state of the art and practice of software development, the treatment of NFRs remains to be a challenge. According to the authors in [118], this challenge is due to the four characteristics of NFRs:

- NFRs are system-level requirements which cannot be assigned directly to individual system components; instead, they need to be planned at the infrastructure-level as a whole, with their design aspects then entrusted to system components.
- NFRs are often interconnected and their realization in a system requires a collective and coordinated behavior of the system components and a system-level design strategy.
- NFRs are application-specific and their fulfillment necessarily requires a specific design approach appropriate to their applications.
- NFRs are not only a design time concern, but most crucially a runtime concern. Satisfying NFRs in a system means designing runtime mechanisms that can maintain the system’s NFRs throughout the execution time.

Based on these characteristics, our future work will also investigate ways to measure or evaluate different NFR approaches.
REFERENCES

Part I: General References

35. Sun, J., Zhao, L., Loucopoulos, P., and Zhou, B. (2013) Qra: A quality requirements analysis approach for service systems. Proceedings of International Conference on Services Computing, Accepted,

Part II: NFRs-approach Specific References


OMG (2009) Umltm profile for modeling quality of service and fault tolerance characteristics and mechanisms specification. formal/2009-04-05


Robertson, J. and Robertson, S. (2000) Volere: Requirements specification template,


Anton, A.I., Bolchini, D., and He, Q. (2003) The use of goals to extract privacy and security requirements from policy statements. *TR-2003-17*


Loucopoulos, P., et al.  A Systematic Classification and Analysis of NFRs


