Non-functional requirements metrics in practice - an empirical document analysis

Olsson, Thomas; Berntsson Svensson, Richard; Regnell, Björn

Published in:
Proceedings-MeReP: Workshop on Measuring Requirements for Project and Product Success

2007

Document Version:
Version created as part of publication process; publisher's layout; not normally made publicly available

Non-functional requirements metrics in practice – an empirical document analysis

Thomas Olsson[1], Richard Berntsson Svensson[2], Björn Regnell[1,2]

[1] Sony Ericsson, Lund, Sweden
Thomas.olsson@sonyericsson.com
http://www.sonyericsson.com

2 Lund University, Sweden
http://serg.telecom.lth.se

Abstract. Non-functional requirements (NFR) complement functional requirements with quality aspects and are a central part of software engineering. This paper presents an empirical case study that characterizes the specification of NFR. The study is performed on a large set comprising 2113 requirements on subcontracted technology platforms in the mobile phone domain. The analysis is qualitative using an emerging coding scheme for investigating of frequency and patterns in NFR specifications. It can be observed that as many as 40% of the requirements are NFR. Standards are commonly referenced in NFR and directly quantified NFR as well as NFR without a concrete metric are common. Results on distribution of NFR in different technical areas are discussed. Two hypotheses are identified for further investigations: (1) different areas of NFR are special in their character and require unique treatment, and (2) if interval and scale patterns are aligned with market value breakpoints and cost barriers then prioritization and scoping can be made more effective.

1 Introduction

It is commonly acknowledged that non-functional requirements (NFR) are an important and difficult part of requirements engineering [5, 6]. Non-functional requirements (also known as quality requirements) complement functional requirements with quality aspects [4]. A characteristic of NFR is that they specify certain quality levels and can hence often be quantified. This is important not only for understanding the requirements [6], but also for prioritization and planning [10].

The aim of the study presented in this paper is to empirically analyze NFR specification in practice and to investigate how NFR metrics are used in an industrial context. The study is performed at Sony Ericsson, one of the leading developers of mobile phones. An in-depth analysis of a single case helps us to understand the details of a specific context, and enables comparison with other similar case studies. To our best knowledge, similar studies on NFR metrics do not yet exist, but future empirical studies on the important area of NFR may hopefully be conducted. Another outcome of this research is the classification approach as such. We demonstrate how a classification can be performed in practice, enabling companies to detail their knowledge of their own requirements.

The case study presented here is a document content analysis study [11], focused on understanding how non-functional requirements are specified, in particular which metrics that are being used. A document analysis is an unobtrusive study of an artefact [11]. A specification consisting of more than 2000 requirements is carefully analyzed in a bottom-up manner. The classification scheme is built up as the research progresses. The study is qualitative in the sense that we do not want to confirm any theory, but rather seek to identify patterns and generate hypotheses for further research.

The research questions that guide the presented work are:

1. What are the characteristics of different types of requirements?
2. How are NFR specified?
3. How are NFR quantified?

The paper is structured as follows: Section 2 provides an account of the case study analysis method with its goals, questions and metrics as well as the coding scheme that is applied in the qualitative data analysis. Section 2 also includes the data analysis with descriptive statistics of distribution of coding categories. Section 3 discusses the implications of the data analysis and summarises lessons learned and hypotheses generated. Section 4 gives an account of related work relevant to NFR specification. Finally, Section 5 concludes the paper.

2 Case study analysis

2.1 Research methodology

This case study is an open-ended and exploratory document analysis [11]. The focus is on understanding how non-functional requirements are specified, in particular which metrics that are being used. A document analysis is an unobtrusive study of an artefact. Analysing the content is a quantified codification of the artefact [11].

The coding scheme and relevant aspects to code are derived as the study progresses. To avoid moulding the characterization to a particular standard or classification, the case study does not use a pre-defined characterization scheme. Instead, the questions and metrics are openly defined, see Table 1.

To collect the metrics, a requirements document is analyzed. A content analysis [11] is performed to codify and quantify the specification. The requirements are analyzed and coded with respect to the different metrics that is collected. The document analysis is performed in the following steps:

1. **A preliminary coding was performed to categorize aspects of interest to be coded in more detail.**
   An overall coding of the entire set of requirements was performed. The goal is to have a first categorization of the requirements into classes of functional and non-functional and to explore which aspects to code further. In sub-sequent steps, the effort is focused on the non-functional requirements.

Table 1. Goals, questions and metrics [2]

Goal	Question	Metric
1. What are the characteristics of different types of requirements?	1. Which types of requirements are present in the requirements document?	1. Type of requirements (functional, non-functional) 2. Distribution across different areas of the specification
	2. How many are there of the different types of requirements?	3. Type of requirements 4. Number of requirements of certain types
	3. Are there areas with more or less NFR than others?	5. Type of requirements 6. Distribution across different areas of the specification
2. How are NFR specified?	4. Which metrics are being used?	7. Direct or indirect quantification 8. Type of metric 9. Distribution of metric usage
	5. How is the usage of standards in the NFR?	10. Name of standard 11. Direct or indirect quantification
3. How are NFR quantified?	6. On which kind of scales are metrics defined?	12. Types of intervals 13. Type of scales

2. **The emerging codes are discussed and consolidated.**
   The overall coding was revised and consolidated. The revision consisted of attaining orthogonal categories and agreeing on the meaning of the categories. The consolidation also consisted of raising the level of confidence in the coding. The subjectively perceived coding confidence varied from "very low" to "very high" in five levels. It was agreed that the confidence should be at least judged "high" to be considered acceptable. The coding was performed by all three researchers and discussed until an agreement was reached, with the intention to increase the coding accuracy.
3. **Detailed coding, first iteration.**
   After having identified which categories to code, a more detailed coding of e.g. domain, scales and class followed. In the first iteration, the goal was to get an initial understanding of the requirements. The emerging codes and groups were analysed to derive a consistent and reliable codification. Not all quality requirements were coded, as the purpose was mainly to derive a suitable coding.
4. **Detailed coding, second iteration.**
   The purpose of the analysis of the second coding iteration was to understand the distribution and nature of the types of NFR. The analysis was performed using card sorting [8, 13]. The card sorting was performed as a group activity among the authors. The emerging codes was then used to code all quality requirements in the document.

The coding was performed in parallel by all three authors. To ensure consistency and reliable results, there was an overlap in the coding among authors and the consolidating steps 2 and 4.

The process of coding and consolidating the classes ran over several months. The effort spent on the case study is in the range of 3-4 person weeks. The first author spent more than half of that effort. The remaining was equally distributed between the two other authors.

2.2 Description of the case

The case study is performed at Sony Ericsson. Sony Ericsson operates in the mobile phone industry, developing several millions phone per years for a wide range of markets and customers. The individual products are developed on a common platform with a product line engineering approach. Hence, also the NFR are specified mainly for the platform and not the individual products. Some parts of the platform are developed by different sub-contractors, some by Sony Ericsson itself.

Mobile phones are embedded real-time system consumer products developed for a mass-market. The platform requirements that are investigated in this study are characterized by a number of aspects relevant to NFR, such as metrics and scale but also usage of standard references. By having a better understanding of the company practices regarding NFR, we support our long-term goal to develop effective support which works in a practical context.

This case study investigates a requirement specification given to a sub contractor of Sony Ericsson. This subcontractor provides mobile platform technology for integration into mobile products. The specification contains areas such as radio, multimedia and network. There are both hardware and software requirements in the specification. The areas range from being pure hardware related to being pure software related on a sliding scale. The specification is focused on enabling technologies, rather than end-user requirements.

In total, the requirements document contains 2113 requirements. There is a mixture of functional and non-functional requirements. The specification is written in natural language English. The document is structured into different areas (headings) and sub areas. The structure and depth of sub areas differ among the areas. A requirement typically consists of 1-5 sentences. The specification is reused over time. New requirements are added and obsolete requirements are deleted. The sub-contractor uses the specification as the basis for a statement of compliance in the negotiation process with Sony Ericsson. The specification has been used over a longer period of times for several generations of platforms. Hence, the requirements have been reviewed and used extensively and are of an appropriate quality.

2.3 Coding scheme

The scheme is used to codify the requirements. The scheme consists of three main requirements types

- Pure Functional requirements (PF) – used for the common understanding of functional requirements.
- Pure Quality requirements (PQ) – these requirements are NFR that do not add functionality, but specifies a quality level on functional requirements or puts requirements on the sub-contractor as such.
- Both functional and quality aspects mixed (F&Q) – this category is used when NFR and functional aspects are combined and intermingled in the same requirement. This also includes cases where the requirement includes references to a (part of a) standard that contains both quality and functional aspects.

As the requirements of class F&Q also contain quality aspects, these requirements are also considered in the detailed analysis of the NFR classes. Therefore, the union of the requirements sets PQ and F&Q are given a general type "Q". The detailed coding is focused on the Q set, see the step description in Section .2.1.

In addition to the general types above, a number of other aspects are coded:

- Standard reference in Quality requirement (SQ) – used to indicate whether requirements of the class Q reference a standard or not. For example, "The MPEG coding standard shall be supported".
- Direct Quality level (DQ) – requirements that are quantified or use a metric. This might be a certain time requirement or sensitivity level, e.g. "640x480 (VGA) resolution shall be supported". It can also be the case that a specific level in a standard is referred, e.g. "GPRS class 1 according to the 3GPP standard shall be supported".
- Indirect Quality level (IQ) – even though it is a non-functional requirement, a metric might not be used. It can for example be a general standard reference or for example a maintainability requirement. The sum of all DQ and IQ requirements are all the Q requirements.

For the DQ requirements, the scale characteristics are also coded as follows:

- Lower/upper bound – a DQ requirement that specifies a one-sided interval, either an upper or a lower bound. For example, there are requirements on phone start-up time (upper bound) and data transfer rates (lower bound)
- Min-Max – a DQ requirement specifying a double-sided interval, i.e. both an upper and a lower bound. This is for example used for sensitivity and accuracy requirements.
- Absolute – a DQ requirement that specified an absolute value, i.e. no interval is used. Display resolution is one example of an absolute DQ.
- Discrete/Continuous – indicates whether the scale discrete or continuous. For certain requirements, e.g. memory, only some values are available. For others, e.g. response time, any value on the scale can be used.

2.4 Data analysis

The overall distribution of requirements can be found in Figure 1 and in Table 2. The data have for confidentiality reasons been made anonymous in terms of which areas the requirements cover. We use the main heading in the specification to separate the document into areas. These areas are used throughout the presentation. There are in total 28 areas. Examples of areas are Audio, Multimedia, User Interface, Java Game Engine and Positioning. These areas represent a top-level partitioning of requirements into sub-domains.

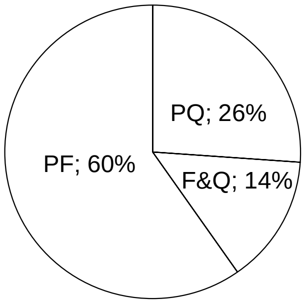

Figure 1 Total distribution of requirements types (functional requirements (PF), non-functional requirements (PQ) and both (F&Q))

As can be seen in Table 2 and Figure 1, functional requirements are the dominating type of requirements. PF requirements represent 60% of the entire requirements mass. 26% of the requirements are pure non-functional requirements (PQ) and the rest (14%) are both functional and non-functional (F&Q). Hence, Q requirements (PQ + F&Q) are 40% of the requirements.

Table 2 Number of requirements per area

Area	PF	PQ	F&Q	Sum	Q	SQ	DQ
1	2	0	0	2	0	0	0
2	0	2	0	2	2	0	2
3	3	1	0	4	1	0	1
4	1	7	0	8	7	0	2
5	8	3	0	11	3	0	2
6	16	1	0	17	1	0	1
7	6	6	6	18	12	0	11
8	7	12	0	19	12	3	2
9	21	3	1	25	4	0	2
10	24	0	4	28	4	4	3
11	2	20	10	32	30	1	22
12	13	12	10	35	22	4	17
13	19	16	1	36	17	11	14
14	22	3	14	39	17	13	0
15	35	3	4	42	7	5	4
16	30	9	3	42	12	8	2
17	32	4	7	43	11	2	5
18	16	8	23	47	31	20	6
19	60	0	3	63	3	0	2
20	40	29	3	72	32	0	12
21	18	53	3	74	56	8	4
22	7	81	2	90	83	40	3
23	63	38	3	104	41	26	25
24	55	37	20	112	57	0	35
25	177	4	40	221	44	14	39
26	198	20	6	224	26	18	11
27	197	42	15	254	57	39	24
28	185	139	125	449	264	97	216
All	1257	553	303	2113	856	313	467

Looking more closely into the different areas in the document, it can be seen that the distribution of the types differs across the document, see Figure 2. Interesting to note is that areas 1 and 2 in the figure have two requirements each and except for area 1, there is at least one Q requirement in the area (cf. Table 2). Note that the number of Q requirements for area 28 is off the chart in Figure 2, Figure 3 and Figure 5. It contains a total of 264 Q requirements. For presentation purposes, the scale is reduced.

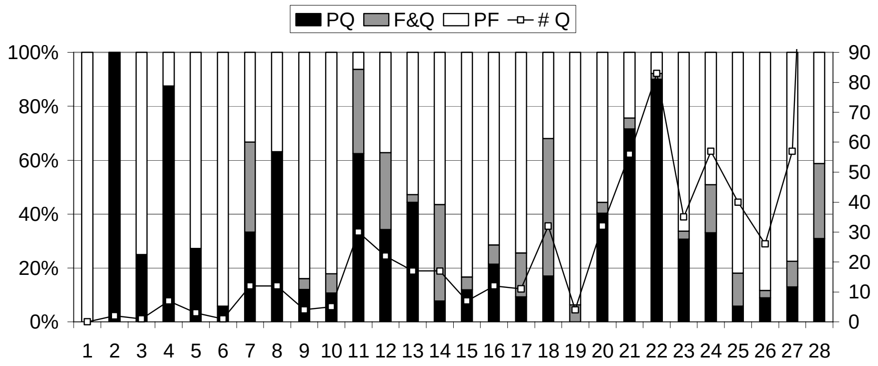

Figure 2 Distribution of types of requirements per area (functional (PF), non-functional (PQ) and both (F&Q)) as well as number of NFR (# Q)

Overall, the mobile phone domain uses a lot of standard references in the requirements specification, see Figure 3. It might both be that there is a general fulfillment requirement, e.g. "the platform must comply with the legal standard XYZ" or a direct reference to a level in the standard "Level 2 of the standard ABC should be fulfilled". The distribution shows that different areas utilized standards in a varying way. 11 areas have no reference to any standard in their NFR. However, 50% of the Q requirements have a standard reference in 9 areas.

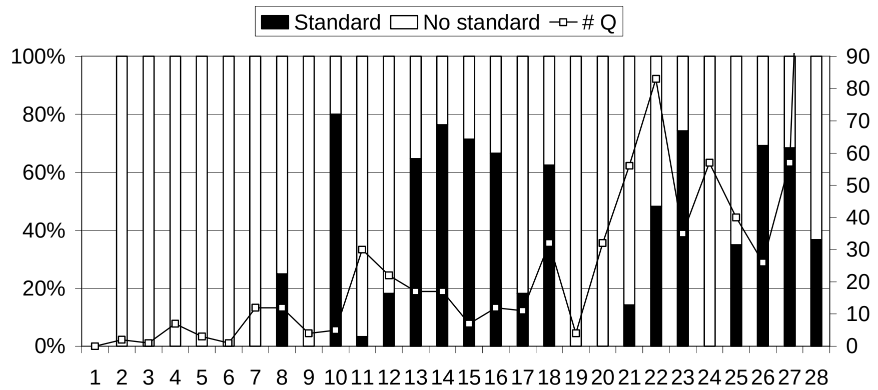

Figure 3 The proportion of Q requirements that reference a standard as well as the number of NFR (# Q)

A non-functional requirement might be specified using a direct quantification (DQ). For example, “The platform should be able to decode MP3 of 128kb/s”. Alternatively, the quality level is indirect (IQ), e.g. “The platform should comply with the 3GPP standard”. The distribution of DQ and IQ requirements can be seen in Figure 4.

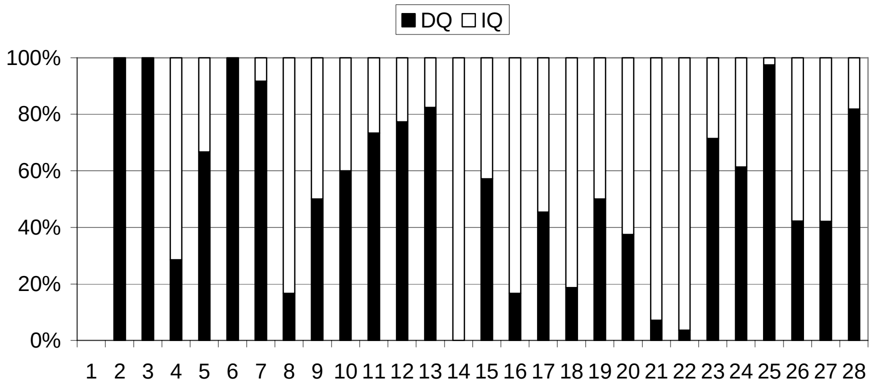

Figure 4 The relative number of Q requirements that have a direct quantification (DQ) or not (IQ)

As with the SQ requirements, it is clear that the different areas are not the same with respect to directly quantified requirements. Most areas have a mixture of IQ and DQ. When combining the standard and metric categories, 49% of the DQ requirements have a standard reference, while 31% of the IQ requirements have a standard reference. Figure 5 depicts how many percent of the requirements have a standard reference. The figure also shows how many Q requirements there are in each area.

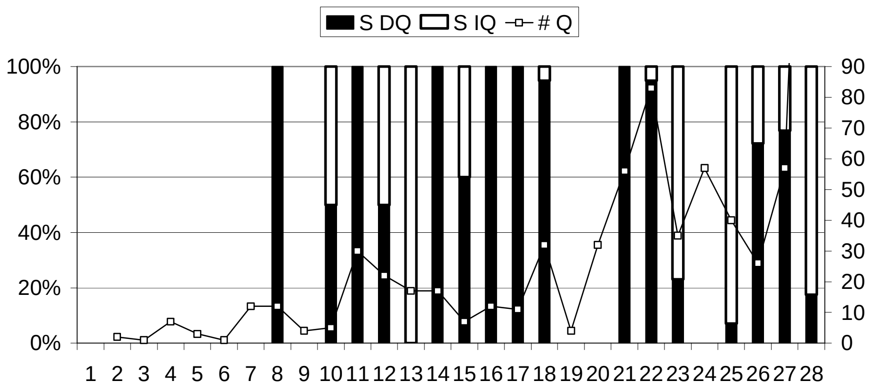

Figure 5 Distribution of standard references, divided on DQ and IQ

Domains of NFR were coded for all the requirements (step 4 in section 2.1). The heading structure (areas) is closely related to domain, but the mapping is many-to-many. For example, Power or radio requirements exist throughout the document. The domain coding is performed from a end-user point of view, while the areas are more technically organized. As can be seen, the number of NFR varies across different domains, see Figure 6.

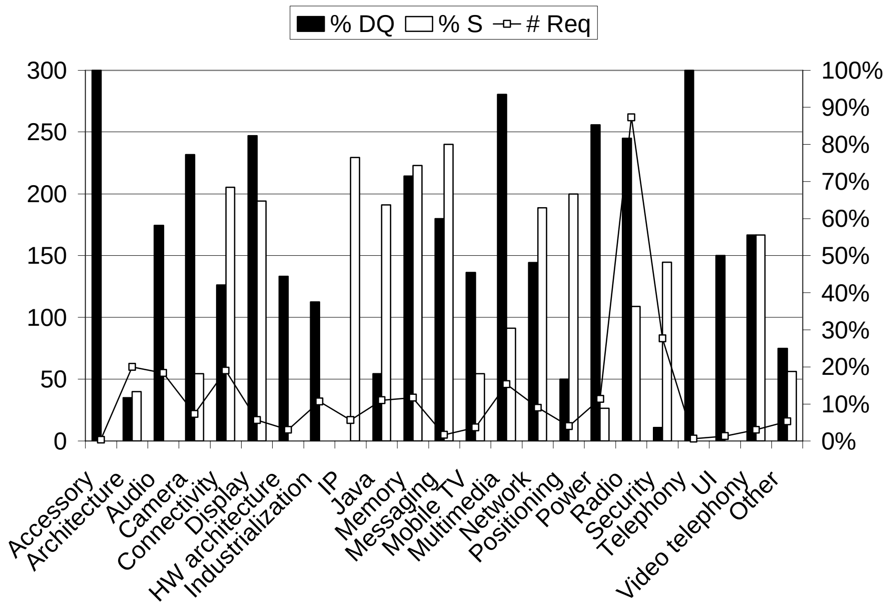

Figure 6 Domains of Q requirements

Finally, scales and types of NFR were coded for all DQ requirements. Looking more closely on the DQ requirements and the scales on which the quantification is specified, a mixture can be seen (see Figure 7):

- Absolute – no interval given, but an absolute number (58%)
- Min-max – a lower and a upper bound is specified, creating a min-max interval (7%)
- Upper bound – a upper bound is specified, creating a one-sided interval (24%)
- Lower bound – as with upper bound, a one-sided interval (12%)

In addition to the interval bounds, the scale might be either discrete or continuous. Memory is an example of a discrete scale. Memories typically come in sizes based on the power of 2, e.g. 64MB or 256MB, and combinations of such sizes. In comparison, e.g. transfer speeds are typically not limited to discrete steps but can assume any value within the available range, e.g. download capacity can vary from 100Kbps to 2Mbps. Area 28 is off the scale with a total of 214 DQ requirements on a continuous scale.

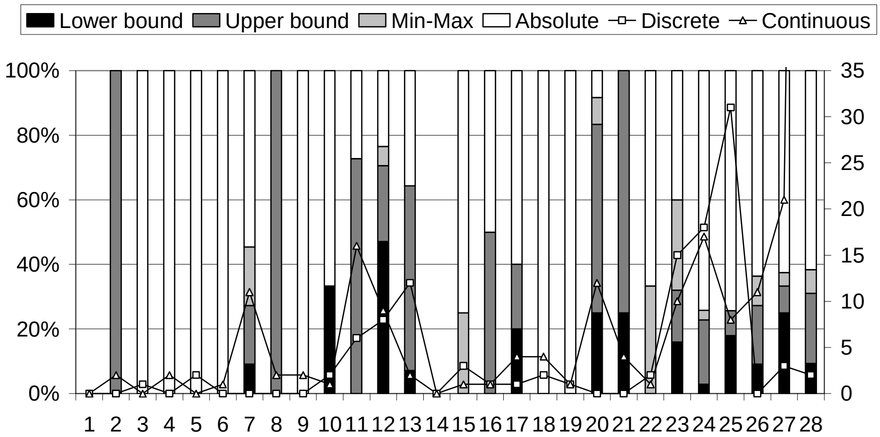

Figure 7 Interval (bars) and scale (line) patterns

3 Discussion of findings

Figure 1 and Figure 2 show the distribution of pure functional, pure quality and mixed functional and non-functional requirements per area. The spread is large ranging from 100% pure functional to 0% pure functional and from 100% pure quality to 0% pure quality. The mixed functional and quality category ranges from 0% up to 83%. This further strengthens the view that the areas are very different and suggest that each area needs unique treatment of its NFR.

Figure 3, Figure 4 and Figure 5 also show vast differences among areas in terms of the fraction of direct metrics and the fraction of NFR that have references to standards external to the requirements specification. This implies even more that the different areas are different in nature and require tailored treatment. Similar observations have been made in [5, 7].

The differences among areas may be explained by the following conjectures:

- The areas are technically very different ranging from having their major emphasis on hardware or mixed hardware and software or mainly software.
- The number of requirements in each group varies greatly, ranging from 2 to 449 with a median of 40 and a mean of 75 requirements per area.
- Some areas are easier to measure than others. Typically, physical measures in areas related to hardware are more straight-forward than subjective experience of software performance.
- Some areas have more maturity than others, in terms of how long requirements belonging to the area have been present in the platform.
- Some areas may have more dedicated resources than others in terms of number of engineers working with the requirements of a particular area.

- Similarly, some areas may have been more thoroughly elicited, specified or validated. This is natural since time is a limited resource and the work needs to be prioritized.
- Some areas may be more critical from a quality viewpoint than others.
- Some areas may be more critical to important stakeholders than others.

This leads us to the first hypothesis: [**H1**] *Different areas of non-functional requirements, both with respect to technical domain and type of non-functional requirement, are unique in their character and require unique treatment in terms of tools support and method guidance.*

More work is needed to identify and classify the different areas and to find patterns within similar areas, across domains. If such patterns exist, these can be used to tailor the support for working with NFR within a certain area. This confirms observations from other case studies [5]. However, the underlying assumptions of the different areas and their unique behaviour are not well known and warrant further studies.

When regarding the distribution of different type, there are many NFR that are related to the architecture or associated documentation in contrast to requirements on the system as such. Performance requirements are the most common type of NFR in this case study, whereas reliability requirements are largely lacking. This leads us to the following observations:

- In the mobile phone domain size and power consumption are central along with limited computer capacity, which can explain why performance requirements are very common. Furthermore, mobile phones are not what are usually considered to be safety critical systems, such as trains or aeroplanes. This explains the limited emphasis on reliability requirements. If a different domain was analyzed, the results would probably have looked different, suggesting that different domains require different solutions.
- There are a number of NFR that are identified as having a concurrency aspect. This leads us to two conjectures: it is not always clear to which type a NFR belongs to and that NFR might belong to more than one type at the same time. Hence, when modelling quality, we need to consider that requirements might be cross-cutting and not possible to sub-divide in a strictly hierarchical manner.
- Many requirements are put on the architecture. In many cases Sony Ericsson has an architecture which needs to be taken into consideration for the sub-contracted parts. If those parts do not fit into the overall system architecture, obvious problems will occur. This can explain why there are so many NFR in architecture related classes.
- Another reason that many requirements are related to the architecture might be that some requirements cannot be revealed to the sub-contractor for confidentiality reasons. Hence, rather than stating the end-user requirements, architecture or design requirements are specified.

Figure 7 is based on coding of direct NFR (DQ) and reveals that there are several different patterns for specifying intervals. It can be seen from the distribution that it is rather common that direct quality metrics are qualified with more information than just a simple absolute value. Furthermore, Figure 7 shows the coding of direct NFR

(DQ) and reveals that there are two principally different scales used; a continuous pattern and a discrete pattern.

Based on the data analysis we make the following conjectures regarding intervals and scale patterns:

- In many cases, there is a threshold. Values up to (or down to) a certain value are all acceptable. One example is transfer rate (lower bound). Another example is start-up time (upper bound).
- Sometimes complex interrelationships exist among different NFR. For example, to maximize the conversation quality, the radio output should be as high as possible. However, the radio output level needs to be kept down to reduce current consumption and electromagnetic field affecting the person using the phone. This leads to a min-max interval.
- It is often difficult to specify an exact value for a certain quality level. Therefore, quality is often specified using an interval to indicate an approximate value within a certain range.
- There are cases where a specific level is sought, e.g. with video resolution. Due to for example compatibility issues, size restrictions and user quality perception, perhaps VGA 30fps is the only acceptable compromise, neither more nor less.
- When referring to a standard, a specific level is sometimes defined. This is then an absolute level in the standard and not an interval. In fact, the scale as such in this case would most likely be nominal and an interval scale would not make sense.
- In certain areas, prominently hardware, the scales are discrete rather than continuous. Hardware components typically have pre-defined steps and combinations which can be used. Hence, we are not free to select for example 34MB of memory. Software, on the other hand, is prominently continuous, e.g. data transfer rate. But there are also examples where scales for software NFR are discrete, e.g. audio and video encoding which is typically standardized for interoperability purposes.

The nature of the intervals and scales for different areas are important when defining the quality levels and when negotiating and prioritizing then. This leads us to a second hypothesis: [**H2**] *Interval and scale patterns need to be aligned with the market value breakpoints and cost barriers [10] to allow effective prioritization and negotiation.*

Each quality indicator needs to be investigated to figure out the optimal way of specifying requirements using intervals. The quality indicators patterns will be different for different areas. To better support prioritization and negotiation (e.g. as in [10]) these patterns need to be understood for the domain. Furthermore, the complex interaction among the different quality indicators and NFR need more study to effectively support negotiation and prioritization.

The coding of NFR is a non-trivial task. It has been reported that when using the IS0 9126 it is inherently difficult to get a reliable classification [2]. Similar problems were identified in this case study. We address this by tailoring the coding scheme to the particular case. This made it easier to attain a reliable coding. The down-side of using a tailored code is that the comparability with other studies is hampered. However, each domain is unique in one way or another and we believe that the coding also needs to be tailored to the domain [5, 7].

There are a number of threats to the validity of the results. First, the coding reliability may in some cases be low due to the deep domain knowledge required but sometimes lacking. This is addressed by independent coding among three researcher and consolidation based on perceived confidence.

Second, the transferability of the result can be discussed as this study only covers one specific case. Preferably, a standardized coding should be used to enable comparison with other cases. To the authors' knowledge, there are as of yet no similar studies performed. Hence, there is no standardized coding and no cases to compare with. Hopefully, the experience and outcome of this study will inspire and aid other researchers in performing similar studies, as there is a lack of empirically grounded understanding of NFR.

4 Related work

Research in non-functional requirements has concentrated on modelling and representation of NFR. However, research related to classification and measurements of NFR are also introduced in literature. In this section, a selection of classification methods and measurements are presented.

There are case studies reporting using different NFR approaches [1, 5, 6, 12]. However, even though industry case studies were sometimes conducted, little or no information is shared on the characteristics of the NFR.

Al-Balushi et al. [1] developed a tool called ElicitO. The purpose of the tool is to empower requirements analysts during elicitation interviews. ElicitO is based on domain ontology to support elicitation activities. The domain ontology uses characteristics and metrics from the standard quality model ISO/IEC 9126 [9].

In the IESE NFR method [7] based on the ISO 9126 [9], quality models are used to document the understanding of quality attributes. The method has shown its usefulness in several industrial case studies [5]. The basis of the method is the concept of quality models. These models capture the behaviour of a specific quality attribute and break it down hierarchically. However, the quality models used in the method is not based on empirical data. Furthermore, detailed support on scales, intervals and interrelationships are largely missing.

According to [12], commercial-off-the-self (COTS) based systems require metrics for quality indicators such as complexity, performance and so forth. Obtaining system-level indicators of quality is complicated for COTS products. Sedigh-Ali et al presents 13 system-level metrics for COTS based systems. How to measure these metrics are described, but no concrete or precise measures are defined. Furthermore, the 13 system-level metrics in [12] are not empirically validated.

Coding of NFR can be a difficult and unreliable process. Al-Kilidar et al [2] empirically evaluated ISO 9126 [9] as a mean to classify NFR. They found both shortcomings in the content of the standard and unreliable codes in an experiment. There is a lack of studies on the matter of coding NFR which needs to be addressed.

5 Conclusion

This paper presents an empirical study on non-functional requirements (NFR) in a requirements specification within the embedded software domain based on a document content analysis of 2113 requirements. As many as 40% of the requirements in the specification are non-functional. The distribution across the areas on the specification varies, but only a few areas completely lack NFR. In many cases (14%), the requirements specify both a functional and a non-functional aspect.

Standards are commonly used and for NFR, as many as 37% of the NFR references a standard. About half of the NFR are specified using a metric. Large parts of the quantified NFR are specified on an interval, both single- and double-sided.

A general conclusion is that for a method to be successful, it is important that it is flexible enough to handle the diverse nature of NFR. This impacts all areas of requirements engineering, starting with elicitation and analysis to specification and validation. Further research is needed into different sub-domains to be able to identify general patterns and trends which can be used to facilitate industry in their work with NFR. This case study results are specific to the domain investigated and to the product platform under study and the results cannot be directly transferred to another context. However, a number of hypotheses of general interest can be stated based on this case study, as discussed in Section 3.

We used a domain-specific method for coding NFR which can be applied instead of basing a classification on a general model such as ISO 9126. Even though it requires an initial tailoring to be useful, once the coding scheme is defined, it is our experience that domain-specific coding can be reasonably reliable and efficient.

To improve how NFR are handled, we need to understand the characteristics of them, and this case study is aimed as a step towards a richer picture of NFR rooted in empirical findings in a specific domain. Further case studies are needed in other domains and on other requirements specifications to enable generalizations outside the domain of this study.

References

1. Al Balushi, T.H., Samoaio, P.R.F., Dabhi, D., Loucopoulos, P.: ElicitO: A Quality Ontology-Guided NFR Elicitation Tool. In: Proc. 13th working Conf. on Requirements Engineering: Foundation for Software Quality. (2007) 306-319
2. Al-Kilidar, H., Cox, K., Kitchenham, B.: The use and usefulness of the ISO/IEC 9126 quality standard. In: Proc. International Symposium on Empirical Software Engineering (2005) 122-128
3. Basili, V.R., Caldiera, C., Rombach, H.D.: Goal Question Metric Paradigm. In: Marciniak, J.J. (eds.): Encyclopedia of Software Engineering, Vol. 1. John Wiley & Sons (1994), 528-532
4. Chung, L., Nixon, B.A., Yu, E., Mylopoulos, J.: NFR in Software Engineering. Kluwer Academic Publishers (2000)

5. Doerr, J., Kerkow, D., Koenig, T., Olsson, T., Suzuki, T.: Non-functional requirements in industry – three case studies adopting an experience-based NFR method. In: Proc. 13th IEEE International Conference on Requirements Engineering. Paris, France (2005)373-382
6. Jacobs, S.: Introducing Measurable Non-functional requirements: A Case Study. In: Proc. 4th IEEE International Symposium on Requirements Engineering. Limerick, Ireland, (1999) 172-179
7. Kerkow, B., Doerr, J., Paech, B., Olsson, T., Koenig, T.:, Elicitation and Documentation of Non-Functional Requirements for Sociotechnical Systems. In Maté J. L., Silva, A. (eds.): Requirements Engineering for Sociotechnical Systems. Hersey: Information Science Publishing, (2005) 284-302
8. Nurmuliani, N., Zowghi, D., Williams, S.P.: Using Card Sorting Technique to Classify Requirements Change. In: Proc. of the 12th IEEE International Requirements Engineering Conference (2004)
9. ISO/IEC 9126-1:2001(E): Software Engineering - Product Quality - Part 1: Quality Model (2001)
10. Regnell, B., Höst, M., Berntsson Svensson, R.: A Quality Performance Model for Cost-Benefit Analysis of Non-Functional Requirements Applied to the Mobile Handset Domain. In: Proc. Working conference on Requirements Engineering, Foundation for Software Quality (2007)
11. Robson, C.: Real World Research, Blackwell, Oxford (2002)
12. Sedigh-Ali, S., Ghafoor, A., Paul, R.A.: Software Engineering for COTS-Based Systems. IEEE Computer, 34:5 (2001) 44-50
13. Upchurch, L., Rugg, G., Kitchenham, B.: Using Card Sorts to Elicit Web Page Quality Attributes. IEEE Software, 18:4 (2001) 84 - 89

Acknowledgements. This work was partly funded by VINNOVA (Swedish Agency for Innovation Systems) within the MARS project.