An investigation of how quality requirements are specified in industrial practice

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

Published in:
Information and Software Technology

DOI:
10.1016/j.infsof.2013.01.006

2013

Information and Software Technology 55 (2013) 1224–1236

Contents lists available at SciVerse ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

An investigation of how quality requirements are specified in industrial practice

Richard Berntsson Svensson *, Thomas Olsson, Björn Regnell

Dept. of Computer Science, Lund University, Box 118, 221 00 Lund, Sweden

ARTICLE INFO

Article history:
Received 17 January 2012
Received in revised form 14 January 2013
Accepted 21 January 2013
Available online 7 February 2013

Keywords:
Quality requirements
Case study
Qualitative research
Document analysis
Metrics

ABSTRACT

Context: This paper analyses a sub-contractor specification in the mobile handset domain.
Objective: The objective is to understand how quality requirements are specified and which types of requirements exist in a requirements specification from industry.
Method: The case study is performed in the mobile handset domain, where a requirements specification was analyzed by categorizing and characterizing the pertaining requirements.
Results: The requirements specification is written in structured natural language with unique identifiers for the requirements. Of the 2178 requirements, 827 (38%) are quality requirements. Of the quality requirements, 56% are quantified, i.e., having a direct metric in the requirement. The variation across the different sub-domains within the requirements specification is large.
Conclusion: The findings from this study suggest that methods for quality requirements need to encompass many aspects to comprehensively support working with quality requirements. Solely focusing on, for example, quantification of quality requirements might overlook important requirements since there are many quality requirements in the studied specification where quantification is not appropriate.

1. Introduction

Software has become a substantial part of both industrial and consumer products, and as a consequence, the complexity of the software has escalated. Hence, requirements engineering (RE) is a cornerstone in software development, and central for success [1]. A software product's characteristics are determined by both functional requirements (FRs) and quality requirements (also called non-functional requirements) [2]. A FR specifies *what* the system should perform, while quality requirements (QRs) specify how well it should be performed [2], for example, "*it shall not take longer than 1 second to open the web browser application*".

To increase the chance of market success, it is important not only to develop a software product that meets customers' requirements and expectations, but also offers high value for the software development company as well as for the customers. Hence QR are a key concern throughout the software lifecycle [3,4]. Therefore, QR can be seen as a key competitive advantage [5]. However, despite the importance of QR, it is generally acknowledged that QR are difficult to capture and specify. Several studies, e.g., [2,7,6,8–11] have identified challenges of QR as: difficult to elicit, often poorly understood, generally stated informally in a non-quantifiable manner, where QR should be documented, and difficulties to get attention for QR.

If methods for managing QR are considered immature or even unusable by industrial practitioners e.g. due to problems with scalability [12], they are not likely to advance current practice. A first step towards developing effective and efficient methods for QR is to understand in more detail the problems faced in industry. The importance of well specified and quantified QR have been recognized in the literature. For example, Berntsson Svensson et al. discovered that the difficulties in prioritizing QR are related to, e.g., well specified and quantified QR [12], while Jacobs reported that the introduction of a new method with focus on QR and quantification of QR, enabled test cases to be based on measurements instead of being untestable [10].

However, to the best of our knowledge, no study has actually looked into a requirements specification in industry to analyze how QR are specified, in particular how QR are quantified, and which types of requirements exists in a requirements specification from industry. This paper presents the results of a case study that includes data collected through a requirements specification that consists of 2178 requirements from a market-driven development case company. After an early analysis of the requirements specification, a short paper [13] was presented at a workshop. This paper extends our previous report on preliminary findings [13] with more in-depth description of the requirements specification and account of research methodology, as well as a more thorough analysis, discussion, conclusions, and examples of requirements. The study focuses on understanding QR and how they are specified, in particular how metrics are used in an industrial requirements specification within a market-driven company developing

* Corresponding author. Tel.: +46 46 222 0368; fax: +46 46 13 10 21.
E-mail addresses: richard.berntsson_svensson@cs.lth.se (R. Berntsson Svensson), thomas.olsson@cs.lth.se (T. Olsson), bjorn.regnell@cs.lth.se (B. Regnell).

0950-5849/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.infsof.2013.01.006

embedded systems. The goal is *not* to test a specific theory or treatment, but to understand a specific phenomenon, namely the requirements specification of QR in a real world industrial situation. Hence, the chosen research approach is open-ended, exploratory, and qualitative research [14].

The remainder of this paper is organized as follows. In Section 2, the background and related work are presented. The case company is presented in Section 3, while the research methodology is described in Section 4. Section 5 presents the results and relates the findings to previous studies, and Section 6 gives a summary of the main conclusions.

2. Related work

Research in the area of QR has concentrated on modeling and representation of QR. However, research related to specification, classification, and measurement of QR are also introduced in the literature.

Borg et al. investigated the management of QR in two development organizations [7]. The results show that QR are discovered too late, or not discovered at all; difficulties in prioritization of QR; and difficulties to estimate the cost of implementing QR, and to quantify QR. In another study, Grimshaw and Draper conducted four case studies with an attempt to focus on the QR determination process and improve the understanding of structured methodologies in that process [15]. Grimshaw and Draper found that QR are often overlooked, methodologies for QR do not help in the elicitation process, and there is a lack of consensus about QR [15]. Lubars et al. conducted a field study on requirements modeling and found that the rationale of performance requirements is not always obvious, and that usability requirements should not be documented in the requirements specification [16]. In another survey, Kamsties et al. found that requirements are too vague to test, and challenges related to specification of usability requirements were identified [17]. In Berntsson Svensson et al., three important challenges of QR were highlighted: (1) how to get QR into the projects when FR are prioritized, (2) how to know when the quality level is good enough (i.e., should, e.g. the performance be 2.0 s, 2.2 s, or 1.5 s), and (3) how to achieve testable QR [18]. Moreover, in another study by Berntsson Svensson et al., the results show that QR are sometimes specified in a quantifiable manner [6].

In the literature there are several suggestions of how QR should be elicited. Cysneiros and Leite argue that QR should not be dealt within the scope of FR because QR require a more detailed reasoning [19]. On the other hand, Doerr et al. argue that the elicitation of QR, FR, and the architecture must be intertwined because the refinement of QR is not possible without detailed FR and architecture [20]. In addition, Hassenzhal et al. argue that it is important to gather different aspects such as QR, design approach, and the relationships among them to ensure a basic understanding of the design problem [21].

Several studies have addressed the perceived importance of different types of QR. Johansson et al. found that reliability was identified by a multitude of stakeholders to be the most important QR [22], which was also identified as the most important QR for intranet applications by Leung [23]. Sibisi and Waveren [24] reported functionality as the most important one for two projects, while [25] identified usability requirements as the top QR type. In Berntsson Svensson et al., types of QR were studied from two perspectives, business-to-business (B2B) and business-to-consumer (B2C) [6]. The results show that safety and performance requirements are the two most important QR for B2B companies, while usability and performance requirements were the most important ones for B2C companies.

In the literature, only two methods for specifying measurable QR have been empirically evaluated [26], the Gilb style method [10] and the QUPER model [27]. Jacobs introduced and evaluated the Gilb style method at a case company [10]. To make QR measurable, concepts such as scale (the unit in which the requirement should be measured) and meter (how the measurement will be performed) were used. The method puts a focus on QR and a common understanding of QR was considered as crucial. By using the concept of meter, it was found that test cases were already defined during the RE phase. The QUPER model [27] has two main concepts: *breakpoints* and *barriers*. A breakpoint is an important aspect of the non-linear relation between quality and benefit, while barriers represent an interesting aspect of the non-linear relation between quality and cost. The two concepts of breakpoints and barriers provide three views: (1) the *benefit view*, (2) the *cost view*, and (3) the *roadmap view*. Quality indicators are identified to measure the aspects of quality of interests, where a level of benefit offered by competitors are looked at. This information is used to plan needed level of quality for future release of the software product.

Al-Kilidar et al. evaluated the ISO/IEC 9126 [28] standard in terms of its ability to quantify and measure the quality attributes of a software design [29]. The results show that the "common language" proposed by ISO/IEC9126 did not have a standard interpretation. The authors argue that ISO/IEC 9126, in its present form, does not achieve any of its objectives. Moreover, Berntsson Svensson et al. found that there may be a possible mismatch between the established academic interpretation of quality characteristics of ISO/IEC9126 and the industrial interpretation of it [18].

3. Case company description

The case study is conducted at a case company that develops software and hardware for the mobile handset market. The case company has more than 5000 employees and develops their products, about 20–40 unique mobile phone models each year, for a global and competitive market where several millions of phones are sold each year. The individual products are developed on a common platform using a product line approach [30]. Hence, QR are mainly specified for the platform instead of for individual products. The case company has several consecutive releases of a platform (a common code base of the product line) where each of them is the basis for one or more products that reuse the platform's functionality and qualities. The case company has two types of platform releases, a *major* and a *minor* release. A major release has a lead-time between two and three years from start to launch, and the focus is on functionality growth and quality improvements of the product portfolio. Minor platform releases usually focus on the platform's adaptations to different products. Various sub-contractors develop parts of the platform.

This case study investigates a requirement specification, which is described in the following subsection, given to a sub-contractor of the case company. This particular sub-contractor provides mobile platform technology for integration into mobile products. However, the sub-contractor does not only provide mobile platforms to the case company, they provide platforms to the case company, but to their competitors as well.

3.1. The requirements specification

The sub-contractor specification contains requirements in different sub-domains, which in practice can be seen as a collection of several independent specifications for different sub-domains. The different sub-domains, which range from being very hardware centric to pure software sub-domains with several experts in each sub-domain and little overlap of the expertise across sub-domains, are presented in Table 7 in Appendix A. Hence, the different sub-domains can more or less be viewed as independent specifications, written by different practitioners at different points in time and with different ways of specifying requirements. In total, the requirements specification contains 2178 requirements, including both hardware and software requirements, as well as functional and quality requirements. The requirements specification is written in English using natural language where a typical requirement consists of 1–5 sentences, and the requirements specification is structured hierarchically. The QR range from being pure hardware to pure software related. The requirements specification is focused on enabling technologies rather than end-user requirements as the specification in question is for the core platform, and not for end-user applications. The requirements specification is reused over time. New requirements have continuously been added, while obsolete requirements have been removed.

The sub-contractor uses the specification as the basis for a statement of compliance in the negotiation process with the case company. The specification has been used over a longer period of time for several generations of platforms. Hence, the requirements have been reviewed and used extensively over the years and across several releases. Furthermore, sub-domain experts, usually between 2–10 experts in each sub-domain, write the different requirements that are associated with each sub-domain.

4. Research methodology

The investigation presented in this paper was carried out using a qualitative research approach, namely a case study [31]. Qualitative research aims to investigate and understand phenomena within its real life context [32]. A qualitative research approach is useful when the purpose is to explore an area of interest, and when the aim is to improve the understanding of phenomena [31,32]. The purpose of this study is to gain in-depth understanding of how QR are specified, in particular, how metrics are used in an industrial requirements specification within market-driven embedded system companies. Due to the focus of this study, an exploratory case study methodology was chosen since case studies are an in-depth investigation of phenomena on a specific case. In addition, in the case of an exploratory case study, little knowledge about the phenomena is available; hence the study aims at identifying propositions and hypotheses, which can be used in forthcoming confirmative research and empirical studies, such as case studies, of QR. Our approach was to explore how QR are specified without preconceived hypotheses aiming for an unbiased understanding of the case [14]. The research questions in Table 1 provided the focus for the empirical investigation.

As the focus of this study is how QR are specified, a requirements specification (archival data [31]) was analyzed in depth, also called content analysis [32]. Content analysis is a method for analyzing and interpreting data [14,32]. The focus of content analysis is to gather information and generate findings. The gathered information (content) can be any written information and different categories containing content are constructed for analysis. After the content has been gathered and categories been constructed, it is analyzed and conclusions based on the content is reported. All three authors examined the categories, first individually and then together in a workshop setting. The category analysis included examination of the content from different perspectives. One advantage of using archival data is that access to the authors of the requirements specification is not needed. Hence, the analysis in this study is based on the requirements specification alone, along with the experience of the researchers.

Table 1
Research questions.

Research questions (RQ)
RQ1: How are quality requirements distributed in a requirements specification?
RQ2: How are quality requirements specified, especially how are they quantified?
RQ3: What different types of quality requirements exist in a requirements specification?

4.1. Data collection and analysis

Since an exploratory methodology [32] was used in this research, no pre-defined categories were used during data collection and analysis. At the start of the analysis, much focus and attention were given to the development of a good categorization of the requirements. We started out without a defined set of categories since we did not want to limit the categorization in the beginning. Therefore, the first steps of the categorization were used to come up with a set of suitable categorize. The categorization of requirements was conducted in four steps. In the following, each step is described in detail.

1. *A preliminary categorization was performed to identify categorize of interest to be analyzed in more detail.*
   An overall categorization of the entire set of requirements was performed. The goal was to have a first categorization of the requirements into classes of functional and quality requirements, and to explore a detailed classification. In this step, all requirements were considered, not just QR. In the subsequent steps, the effort is focused on the QR.
2. *The emerging categories are discussed and consolidated.*
   The overall categorization 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 categories. The subjectively perceived categorization 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 categorization was performed by all three researchers and discussed until an agreement was reached, so called observer triangulation [32].
3. *Detailed categorization, initial iteration.*
   After the identification of which categorize to collect in the second step, a more detailed categorization of sub-domains, scales and characteristics of the QR followed. During the first iteration, the main goal was to get a first understanding of the QR. The requirements categorization was performed (by all authors) on different parts of the requirements specification in a random manner. Then, the emerging categories and attributes (e.g. scale and interval) were analyzed to derive a consistent and reliable categorization. However, not all QR were categorized in this step, as the purpose of step 3 was to derive a suitable and consistent categorization of the detailed classification. The categorization of all requirements was conducted in step 4. Similar to step 2, observer triangulation was used to limit the influences of the individual researcher on the requirements specification.
4. *Final detailed categorization.*
   The purpose of step 4, the final detailed categorization iteration, was to categorize all of the existing QR in the requirements specification. This categorization provides the final result presented in Section 5. As in steps 2 and 3, observer triangulation was used to make sure the categories were consolidated and consistent before finalizing the last step.

In all four steps, the categorization was performed in parallel by all three authors. In addition to the consolidation in the last three steps to ensure consistent and reliable results, there was an overlap of the categorization among the three authors. As the four steps describe, the categories and the attributes were built up as the study progressed, and instead of "forcing" a requirement into a category and avoiding categorizing an aspect previously not perceived, the categorization scheme was updated to ensure that as many relevant suitable attributes and categories as possible were discovered. Step 2 was mainly a learning step for the authors where much of the time was spent on gaining a common understanding, and a good basis for the continuing of the study.

In the data analysis phase, content analysis [32] and descriptive statistics [31], were used to identify patterns and interesting phenomena, and complemented with examples of requirements to provide further illustrations and background. Given the amount of requirements and number of ways to dissect it, the analysis was performed iteratively over a period of time.

4.2. Validity

In this section, threats to validity in relation to the research design and data collection are discussed. We consider the four perspectives of validity and threats as presented in Runeson et al. [33].

Construct validity regards if the measurements of a study correlate with the constructs of its research questions. In this study we measure the number of FR and QR, the share of quantification of QR, and the distribution over different QR categories. This threat is addressed by observer triangulation. Three researchers have independently classified all requirements, and most of them are classified by at least two researchers. All three researchers reviewed both the overlaps and the uniquely categorized parts to ensure that the classifications are correct, accurate and reliable.

Reliability is concerned with to what extent the data and the analysis are dependent on the specific researchers. This issue is addressed by having a mix of researchers (a practitioner, a senior researcher and a junior researcher). An audit trail (research notes) was kept, to enable the researchers to review the categorization process. All interpretations were reviewed by all researchers to identify non-uniform interpretation. All results are linked to the underlying data to provide traceability and justification of interpretations. The threat concerned with if the same result would be found if re-doing the study, is mitigated by providing a detailed description of the process of categorizing the requirements.

Internal validity concerns whether causal conclusions of a study are warranted or if overlooked phenomena are involved in the causation. This study mainly describes the nature of artifacts of a case and we do aim to find case-effect relations among constructs.

External validity is concerned with the ability to generalize the results and to what extent the findings can be transferred to other cases. This study is a single-case study and its aim is not to make statistically valid conclusions outside this case, but to study this particular case in-depth to understand and describe selected aspects in relation to the case context. The transferability of the results needs to be assessed by comparing this case with other cases in future work. To support transferability we have included case-specific characterizations of the context and system domain (in consideration of confidentiality).

5. Results and analysis

The following three sub-sections present and discuss one research question each, corresponding to the research questions in Table 1.

5.1. Distribution of quality requirements (RQ1)

The requirements specification was analyzed and categorized in detail, and the emerging categories (types and characteristics), which are illustrated in Fig. 1, are a result of a long process. In Fig. 1, we see that a *requirement* (R) can be one of the types *functional* (FR) or *quality* (QR). Since the focus of this study is QR, FR are not further broken down or analyzed.

A requirement has two characteristics, *sub-domain* and *standard* (see Fig. 1). Sub-domain is a grouping of requirements into sub-domains of applications, which are detailed in Table 7 in Appendix A, for example, network access and multimedia. The characteristic *standard* is a tagging whether or not the requirement is directly, or indirectly referring to a specific standard. For example, to the 3GPP standard, which is commonly used in the mobile handset domain, or to a multimedia standard such as video encoding.

A quality requirement has an ISO 9126 characteristic. This characteristic refers to the standard ISO 9126 [28] and is a mapping of the QR to the ISO 9126 standard characteristics (Fig. 6, in Section 5.3, shows the ISO 9126 characteristics found in the requirements specification). Although the ISO 9126 standard has been replaced by ISO 25030 [34], ISO 9126 was chosen as the characteristics because of three reasons, (1) the quality model of ISO 25030 is based on the ISO 9126, (2) ISO 9126 is more widespread in industry, and most importantly, (3) the ISO 9126 standard is currently used at the case company.

A quality requirement is further detailed into two types, *quantified quality requirements* (QQR) and *non-quantified quality requirements* (NQQR). The QQR type is a quality requirement with a direct quantification within the requirement, while a NQQR is a quality requirement without metrics. A QQR has two characteristics:

- *Scale* – whether on a discrete or a continuous scale. Memories, for example, are only available on a 2-multiple scale, e.g., 32 or 64 MB. Typically, 23 MB of memory do not exist. Hence, memory size is a discrete scale. On the other hand, response time is typically on a continuous scale, for example, 4.3 s or 22 ms.
- *Interval* – whether the metric is specified as one value, or as an interval (one-sided or double-sided). A QR such as "Support for encoding frame rate of 15 fps." does not specify an interval; it is an absolute value. On the other hand, "The platform shall support an online 90° image frame rate at minimum 15 fps." is a QQR with a one-sided interval, and "The frame rate change shall be variable between 15 and five (5) fps." is a double-sided interval.

In total, the requirements specification contains 2178 requirements distributed over 20 sub-domains, where the number of requirements per sub-domain varies from 10 to 447, which is illus-

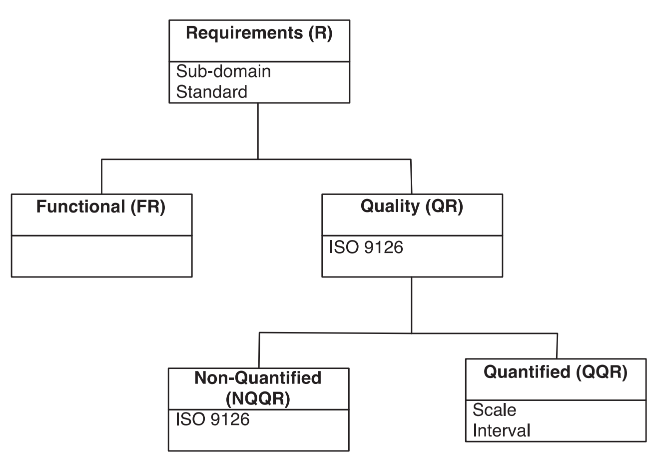

Fig. 1. Emerging categories with characteristics.

trated in Table 2 (FR & QR is a requirement that have both a functional as well as a quality aspect, see Section 5.2). The majority of the requirements are FR (62%), while 38% of the requirements contain an aspect of quality, thus categorized as QR (total QR in Table 2).

Looking at the distribution of types of requirements across the different sub-domains, the results show that no sub-domain completely lacks QR (see Fig. 2). The median is that 35% of the requirements in each sub-domain are QR; however it varies across the sub-domains. The sub-domain Network has the least percentage (13%) of QR, while the Security sub-domain has the highest percentage of QR (92%).

An interesting finding from the categorization process is that the use of standards is common. Standards are found in as many as 38% (in 313 out of 827 QR) of the requirements. An example of a how requirement with a reference to a standard can be specified:

Example 1. Support for AMR-WB.

The support for a specific codec is a FR; however, the standard may contain several other requirements, including both FR and QR. Another example of how a requirement can be specified:

Example 2. Support for H.263 Profile 0, Level 10.

In Example 2, a specific quality level is explicitly pointed out; hence it is a QQR. In addition, the support for H.263 is specified in another (functional) requirement. A third example of how a requirement (NQQR) with a reference to a standard can be specified:

Example 3. The platform shall be R99 compliant.

Although the requirement in Example 3 is only one line, the R99 standard is huge and has large implications on the product. The sub-domains that use most standard references are (see Table 3), Messaging (80%), IMS (76%), and Memory (74%), while the sub-domains with least references to a standard are, HW architecture (0%), Audio (0%), Industrialization (0%), and UI (0%).

5.1.1. Discussion

In the analyzed requirements specification, 38% of all requirements (827 out of 2178 requirements) are QR, but the variation across the sub-domains is large. The variation is related to number of QR (from 4 to 262 QR), how QR are specified, and the types of QR that exists. The differences among the sub-domains may be explained by (1) the technical differences between the sub-domains, ranging from having a main emphasis on hardware, mixed hardware and software, to mainly focusing on software, (2) level of maturity, i.e., for how long requirements for a particular sub-domain have been present in the software product, (3) available resources in terms of practitioners working with requirements, and (4) that some sub-domains may be more critical from a quality viewpoint and more critical to important stakeholders than other sub-domains. Despite the differences across the sub-domains, it seems unlikely that it stems solely from their nature. It seems as if there is a methodological problem as well, with an insufficient support for working with QR. Especially since several sub-domains are deficient on QR, or have a low quantification penetration for QR that should be quantified.

The result that some sub-domains have many QR, while others scarcely any suggests that the priority between FR and QR varies, which is similar to the results in [6]. One predominant problem with QR that may explain the difference of priorities between FR and QR across the sub-domains is their propensity to impact larger parts of a system, span over several (or almost all) FR. That is, some sub-domains may not have the knowledge and understanding of how to manage QR in practice, hence QR have a lower priority. Another possible explanation may be the understanding that FR represents new development that may have higher status than, e.g. improving the performance of the system. In our results, 827 QR (38% of all requirements) have been discovered and specified, which suggests that many QR can, and have been elicited and specified. This result is neither inline with the results in [7] where QR were found difficult to discover, if discovered at all, nor in line with [15] that reports that QR are often overlooked.

Looking into the representation of QR, although the majority of the QR are separated from FR, which is partly in line with [8] who states that QR are usually separated from FR, 36% of all QR (294 out of 827) are requirements with both a functional aspect, as well as a quality aspect (see Example 5). Moreover, according to [8] and to the IEEE Standard 830 – Recommended Practice for Software Requirements Specifications [35], QR are listed separately under different sections in the requirements specification, which is not in line with our results where most QR and FR are grouped into sub-domains based on different areas. In addition, the result that

Table 2
Distribution of requirements across the sub-domains.

Sub-domain	FR and QR	FR	QR	Grand total	Total QR (FR and QR + QR)
Architecture	6	199	59	264	65
Audio	21	57	37	115	58
Camera	10	13	12	35	22
Connectivity	25	196	33	254	58
Display	1	19	16	36	17
HW architecture	0	1	9	10	9
IMS	14	22	3	39	17
Industrialization	2	40	21	63	23
Java	24	16	8	48	32
Memory	3	69	32	104	35
Messaging	2	23	3	28	5
Mobile TV	7	32	4	43	11
Multimedia	32	181	14	227	46
Network	7	181	20	208	27
Positioning	2	31	9	42	11
Power	6	23	27	56	33
Radio	125	185	137	447	262
Security	2	7	81	90	83
UI	1	21	3	25	4
Video telephony	4	35	5	44	9
Grand total	294	1351	533	2178	827

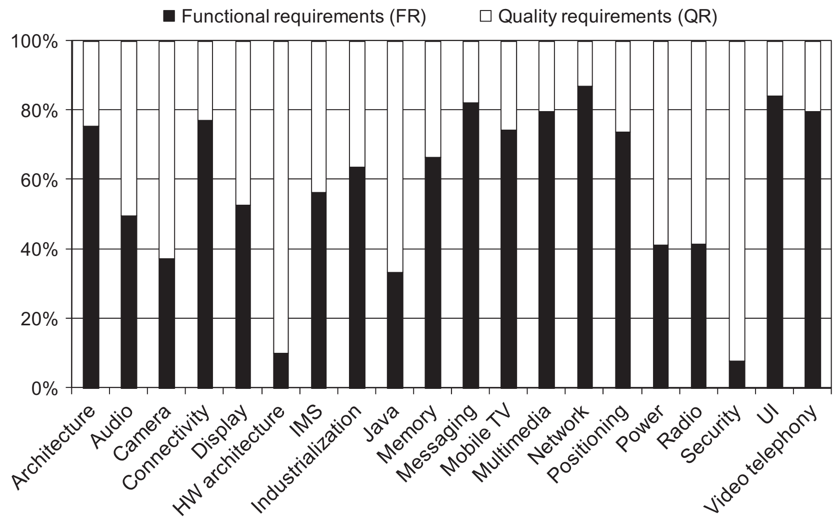

Fig. 2. Distribution of FR and QR across the sub-domains.

Table 3
Use of standard references across the sub-domains.

Sub-domain	No standard	Standard	Total QR
Architecture	54	11	65
Audio	58	0	58
Camera	18	4	22
Connectivity	18	40	58
Display	6	11	17
HW architecture	9	0	9
IMS	4	13	17
Industrialization	23	0	23
Java	11	21	32
Memory	9	26	35
Messaging	1	4	5
Mobile TV	9	2	11
Multimedia	32	14	46
Network	9	18	27
Positioning	4	7	11
Power	31	2	33
Radio	167	95	262
Security	43	40	83
UI	4	0	4
Video telephony	4	5	9
Grand total	514	313	827

36% of all QR are a mix of FR and QR suggests that the elicitation of QR and FR needs to be intertwined. This is in line with the elicitation processes in [19,20], but not in line with Hassenzhal et al. [21] who argues that QR, design approaches, and their relationships should be gathered together, i.e., FR and QR should not be dealt with within the same scope.

The use of standards specifically, but also how much information, such as identification number, QR type, rationale, and originator, a requirement should be comprised of are proposed by some authors, for example [36]. In practice, it is not possible to specify every single detail. However, hiding requirements in standards, or relying on implicit domain knowledge might be risky. It does; however, not appear to change how requirements are written in the analyzed requirements specification. It is not possible to identify any correlation between quantified QR and QR that refers to a standard, or between standards and how requirements are written.

5.2. Specification of quality requirements (RQ2)

The requirements specification is written in structured natural language where requirements are organized by the use of heading hierarchies. In addition, all requirements are numbered with a unique ID. In general, a requirement only contains a single requirement, i.e., two requirements are not written as one. The only exception is the mixing of functional and quality aspects within one requirement. Of all requirements (both FR and QR) in the requirements specification, 14% are a mix of FR and QR, while 24% are "pure" QR. A mixed requirement has both the functional part as well as a quality part included, for example:

Example 4 (*Bluetooth*). Support for multi-link. Clarification: X[1] simultaneous links. (*FR and QQR*)

In Example 4, the functional part is multi-link, while the quality aspect is the number of simultaneous links. Although this requirement is classified as QR, there is a functional aspect, which makes the requirement a mixed requirement. However, for QR or requirements that have both a functional and quality aspect, typically there are never multiple QR within one requirement. One reason for this may be that the general view that each requirement should preferably specify one, and only one function, which may also have been transferred to the specification of QR. Fig. 3 shows the distribution of mixed FR and QR across the sub-domains.

Although two requirements are never written as one requirement, it is not uncommon; however, to have multiple quality level for one FR, for example:

Example 5. "The platform shall receive uncompressed data and shall compress and save the data to desired JPEG size. Clarification: It shall maximum XX s/megapixel to accomplish the whole process for a YX M camera resolution." (*FR and QQR*)

"The platform shall receive uncompressed data and shall compress and save the data to desired JPEG size. Clarification: It shall take maximum XY s/megapixel to accomplish the whole process for a YY M camera resolution." (*FR and QQR*)

In addition, another way of specifying requirements in the requirements specification is to write FR and QR separated:

Example 6 (*Mobile TV*). Support for Time Shift (playback with delay). (*FR*)

The limit for the time shift buffer is available memory. (*NQQR*)

In Example 6, it has already been specified that there should be a time buffer, thus the NQQR specifies the quality level for the time

[1] Actual numbers will not be entered, for confidentiality reasons.

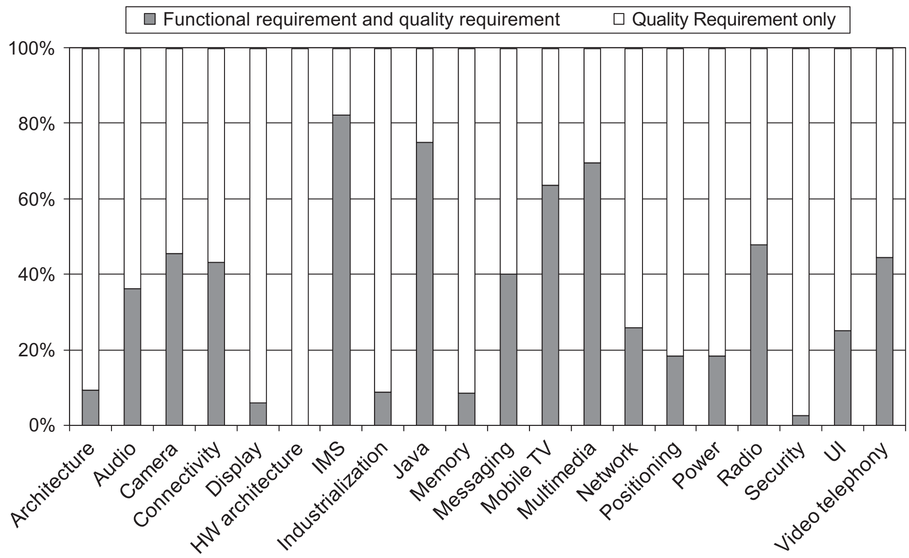

Fig. 3. Distribution of mixed (FR and QR) across the sub-domains.

buffer. A third way of specifying a QR to a particular sub-domain is to use the requirements specification's heading hierarchy. In these instances, the FR is not repeated, but there is one QR for each level of quality. For example:

Example 7. Section 1.2.3. Audio A/D

Support for stereo A/D at 8 kHz. (*QQR*)

Support for mono A/D at 16 kHz. (*QQR*)

Example 7 shows the most common way of specifying QR, i.e., by the use of the heading hierarchy within the requirements specification. By specifying QR in this way, it is made implicit for what the QR refers to.

When looking at the distribution of quantified and non-quantified QR, we see that 56% of all QR are quantified with a direct metric (QQR). However, the variance across the sub-domains is large, which is illustrated in Fig. 4 (the numbers in each bar represents the number of QR in each category, e.g. Architecture has 9 quantified QR and 56 non-quantified QR).

In Fig. 4, the results show that the IMS sub-domain has no QQR at all, and Security has a small portion of QR that are quantified. One reason why IMS has no QQR may be due to many QR with a reference to a standard, which may hide QQR. The Security sub-domain has many specified QR that are not suitable to be quantified, for example:

Example 8. Support for X.509 certificates with MD2_RSA signature.

Furthermore, looking into the distribution of QQR over ISO 9126 characteristics (see Fig. 5), eight ISO 9126 characteristics have no QQR, e.g., Replaceability (sub characteristic of Portability) and Understandability (sub characteristic of Usability), while Security (sub characteristic of Functionality) has few QQR. Among the six main characteristics of the ISO 9126 standard, Efficiency and Reliability have many QQR, while Portability only has a few. Among the sub-domains, the two sub-domains with most QQR are Multimedia and Power (see Example 9 for an example of a QQR).

Example 9. Support for 2500 mA charge current.

QQR can either be specified using a single absolute value (Example 10) or specified using an interval, single (Example 11) or double-sided (Example 12), for example:

Example 10. Support for stereo D/A at 8 kHz. (Absolute, no interval, discrete scale)

Example 11. The maximum delay from call answer is pressed to opened audio paths is XY ms. (One-sided interval, continuous scale)

Example 12. It shall be possible to dedicate a hostbuffer in RAM that is configurable between X to Y MB for HDD. (Double-sided interval, discrete scale)

Looking into how QQR are specified, 57% are specified with an absolute value (like Example 10), 36% with a one-sided interval (like Example 11), while only 7% with a double-sided interval (like Example 12).

The sub-domains with most QQR using an interval are Industrialization and Power. On the other hand, the sub-domains Java and UI only have QQR with absolute values. Looking at Table 4, as many as 77% of the QQR are quantified using a continuous scale; however, there are sub-domains that mainly use a discrete scale to quantify QR, Display (86% of the QQR are on a discrete scale), Multimedia (81%), and Video telephony (80%). Areas that have many quality requirements related to discrete units of information transfer, such as streaming, may result in a natural inclination to include discrete scale quantifications.

5.2.1. Discussion

In the literature, several authors [37–39] have used QR with structured requirements representation notation, for example, combining QR with use cases and misuse cases. However, not a single use case, or misuse case were present in the analyzed requirements specification. One possible explanation may be that some teams may not be used to template-oriented, detailed use cases, but may prefer to use the idea of use cases on a higher level as a gist for a usage context. Instead, all of the requirements in the specification are written in natural language with a heading hierarchy where all requirements are numbered. However, linking or references outside the heading hierarchy is seldom found. This leads to two problematic situations:

- For mixed FR and QR, the functional part is sometimes repeated several times for each quality level, which leads to redundant text and updating problems.
- It is difficult to get an overview of QR associated with a FR.

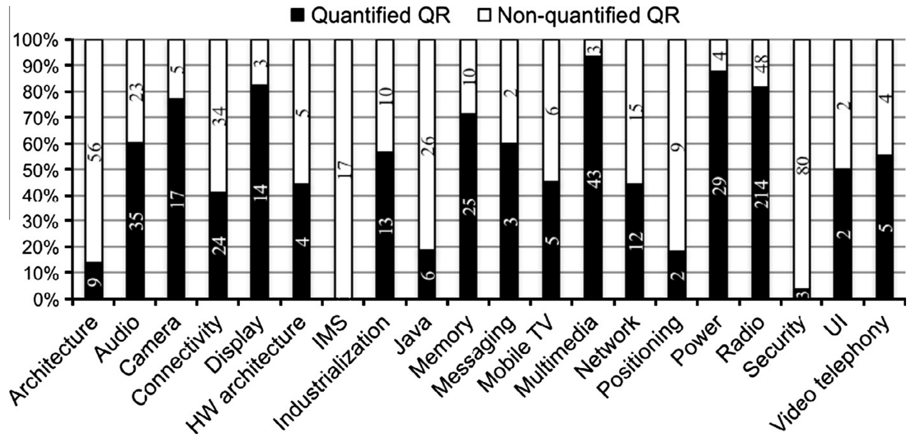

Fig. 4. Distribution of QQR and NQQR across the sub-domains.

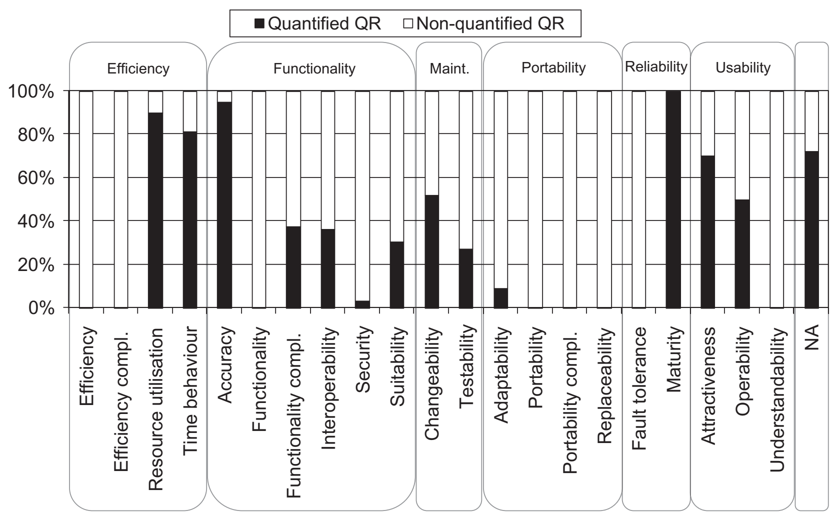

Fig. 5. QQR and NQQR distribution over ISO 9126 characteristics ("NA" represents requirements with no corresponding ISO 9126 characteristic).

The former is a compromise of readability and maintainability of the specification. It is usually easier to read the FR and QR together, but it does tend to clutter the specification by repeating requirements. The latter puts constraints on how relationships between requirements can be expressed in a usable manner. The mixing of QR and FR may be explained by a lack of understanding that the quality part can be considered to be a requirement on its own.

The interdependencies among requirements can cause problems if ignored [6], which makes it difficult to specify crosscutting concerns. Without a structure for specifying interdependencies among requirements, cross-functional aspects might be difficult to specify. As a work-around, there is a separate section on performance in the analyzed requirements specification, which is suggested by [9,35] for example:

Example 13.

2. Performance
...
2.3 FM-Radio
...
2.3.4 FM-radio Record

Listening to FM-radio with Bluetooth headset and record from the radio at the same time.

Table 4
Distribution of QQR using continuous and discrete scales across the sub-domains.

Sub-domain	Continuous	Discrete	Total QQR
Architecture	6	3	9
Audio	17	18	35
Camera	9	8	17
Connectivity	21	3	24
Display	2	12	14
HW architecture	4	0	4
IMS	0	0	0
Industrialization	13	0	13
Java	4	2	6
Memory	10	15	25
Messaging	1	2	3
Mobile TV	4	1	5
Multimedia	8	35	43
Network	11	1	12
Positioning	1	1	2
Power	29	0	29
Radio	212	2	214
Security	1	2	3
UI	2	0	2
Video telephony	1	4	5
Grand total	356	109	465

Use case extension: handling an incoming call (e.g., MP3 ringtone) without stopping the recording.

The problem with this work-around is that not all requirements on a specific subject, e.g. the FM-radio in Example 13, are specified at the same place in the specification. Some requirements on the FM-radio are in the FM-radio section and some are found in the Performance section. The lack of a good overview of all QR related to a FR might lead to missing QR, as the completeness may be more difficult to assess. For QR, the problem becomes even more severe compared to interdependencies between FR since QR are typically crosscutting and affects other QR as well as FR, both in a positive and negative way. By explicitly documenting interdependencies between QR, it becomes easier to see through the crosscutting aspects [6]. In the analyzed requirements specification, both implicit interdependencies exist and cause problems as well as an inadequate structure forcing cumbersome explicit handling, e.g. through repetition of requirements or deep hierarchies. However, it still needs to be proven to be cost-effective before explicit handling of interdependencies can be said to be a general recommendation in industry.

It is sometimes suggested that all QR should be quantified, e.g., in ISO/IEC 9126 [28]. In the analyzed requirements specification, 56% (465 out of 827) of all QR are quantified. The results show that the quantification is given without an interval and a rationale, which is in line with [17]. Without a rationale and an interval, the developers are left guessing why a specified level of quality was chosen, and how to handle even a small deviation from the target when testing even with quantified QR. Furthermore, our study shows that there is a lack of information, such as intervals and rationales, to deduct when the quality level is reached, suggesting that the observations by Berntsson Svensson et al. [18] can be confirmed in this case study. Moreover, Berntsson Svensson et al. found that quality requirements are sometimes specified in a quantifiable manner [6], which is in line with the findings in presented in this paper. Although many QR are quantified in the analyzed requirements specification, several QR are not quantified, and likely should not be quantified.

One possible explanation of why there is a difference across the sub-domains in terms of number of quantified QR may be related to requirements prioritization. It may be easier to prioritize a binary decision of including or excluding a FR compared to prioritizing the level of quality for a QR on a continuous scale, i.e. should the performance level be 1.5 s, 1.3 s or 2.0 s? The sub-domains Radio, Power, and Multimedia have most QQR among the sub-domains. One explanation may be the nature of these sub-domains that makes it easier to quantify QR. For example, Multimedia includes streaming video, encoding and decoding of bit rates, Radio includes radio frequencies, and Power includes charging, battery consumption, which are all rather easy to quantify. Based on the findings, there are many examples in this specification where quantification is not appropriate. Security, for example, is one sub-domain (and ISO 9126 characteristic) that sticks out with a low number of QQR (see Fig. 5). Also areas such as portability and maintainability have few QQR, suggesting that NQQR are also a relevant part of the process and the specification. Therefore, focusing solely on quantifying QR, as in the Gilb style method [10] and the QUPER model [27], is inadequate if applied at the case company. To be able to improve the specification of the QR that should be quantified, a structured and systematic method may help in achieving the quantification.

There are large differences in how the quantified QR are written. It does not always make sense to quantify QR by using an interval (one-sided or double-sided), e.g. sampling frequency. For memories, for example, only specific sizes exist. It is not possible to have 35 MB of memory, it is either 32 or 64 MB. Therefore, depending on the nature of the specific QQR, the quantification will be different. How QQR are specified impacts other parts of the development process in addition to the specification. For example, when prioritizing how much memory there should be in a product in the mobile handset domain (more memory means higher cost) the practitioners are faced with a discrete scale. Moreover, during requirements testing, the use of interval quantification can be vital to assess the outcome of the test case. It is rarely the case that, e.g., 4.2 s, is the only acceptable outcome. Perhaps anything between 4 and 5 s is ok. Hence, using intervals in the QQR clarifies how to interpret the test case outcome. Furthermore, intervals can provide a direction on when, for example, to stop improving the performance. For example, if the target is less than 10 s and the current performance are at 1 s, it is unlikely that further improvements will improve the return on investment. Although better performance is in general better, improving performance beyond certain levels will not increase the value for the end customer [27]. By adding a double-sided QQR, it is clearer when further improvements will not increase the market value.

5.3. Existing types of quality requirements (RQ3)

The domain-specific categorization (see Fig. 1) that was developed for the case company in this study contains five types of requirements, 20 sub-domains (see Table 7 in Appendix A), and information about the quantification of QR (see Section 5.1). The developed sub-domains (see Table 7 in Appendix A) of the specific categorization are comparable to the ISO 9126 sub-characteristic, though they may be on different dimensions. Hence, they are complementary, as opposed to competing. To assign an ISO 9126 characteristic and sub-characteristic to a QR requires both understanding of the QR as well as the ISO 9126 characteristic. The main problem experienced with the ISO 9126 categorization is that often several of the ISO 9126 characteristics are candidates. In the studied requirements specification, ten ISO 9126 sub-characteristics are not present (see Table 5). In addition, an ISO 9126 sub-characteristic could not be determined for some QR. Instead, only the top-level characteristic is assigned.

Fig. 6 shows which ISO 9126 characteristics and sub-characteristics that are present in the different sub-domains, while Table 6 shows the distribution of QQR and NQQR across the ISO/IEC 9126 sub-characteristics. Fig. 6 and Table 6 show that 39% of the QR are in the Efficiency characteristic and 35% in Functionality. In total, 11% of the QR could not be assigned to any ISO 9126 characteristic and is marked "N/A" in Fig. 6 and Table 6.

Looking at Table 6 and Fig. 6, it is not surprising that all sub-characteristics of the Efficiency characteristic are present in the requirement specification, since efficiency is central to the mobile phone. It is a small-embedded system, with an ever increasing amount of uses, pushing the boundaries for that the hardware can deliver. Also, being a telecommunications domain, following standards are central, a major part of the Functionality characteristic. A surprising finding was the lack of maintainability requirements, particular

Table 5
ISO 9126 characteristics not present in the requirements specification.

ISO 9126 characteristic	ISO 9126 sub characteristic
Maintainability	Analyzability
	Stability
	Maintenance compliance
Portability	Replaceability
	Installability
	Co-existence
Reliability	Recoverability
	Reliability compliance
Usability	Learnability
	Usability compliance

Table 6
Distribution of QQR and NQQR across ISO/IEC 9126 characteristics.

ISO/IEC 9126 sub-characteristics	QQR	NQQR	QR
Efficiency	0	5	5
Efficiency compliance	0	1	1
Resource utilization	70	8	78
Time behavior	194	45	239
Efficiency total	264	59	323
Accuracy	37	2	39
Functionality	0	4	4
Functionality compliance	32	54	86
Interoperability	4	7	11
Security	3	90	93
Suitability	16	37	53
Functionality total	92	194	286
Changeability	15	14	29
Testability	7	19	26
Maintainability total	22	33	55
NA	67	26	93
NA total	67	26	93
Adaptability	2	21	23
Portability	0	1	1
Portability compliance	0	3	3
Replaceability	0	7	7
Portability total	2	32	34
Fault tolerance	0	4	4
Maturity	1	0	1
Reliability total	1	5	5
Attractiveness	7	3	10
Operability	10	10	20
Understandability	0	1	1
Usability total	17	14	31
Grand total	465	362	827

since the development methodology at the case company follows a platform principle; hence reuse of the same platform for many products.

5.3.1. Discussion

To assess a requirements specification for quality of the QR, a detailed analysis is needed. As seen in Sections 5.1 and 5.2, there are large differences across the requirements specification. Therefore, a standard set of categories is likely to be too general to be useful, not only for practitioners, but also for researchers. Using a generic standard such as the ISO/IEC 9126 does have drawbacks. As the requirements specification is not written with the standard in mind, many QR can be classified as several characteristics and sub-characteristics. Hence, for the sake of the categorization, the characteristic that all authors felt was the most appropriate was chosen. This indicates that a tailored model may be more appropriate than the use of a standard model, which is also found in [20,29,40]. Furthermore, the "common language" proposed by ISO/IEC 9126 do not have a standard interpretation, hence, ISO/IEC 9126 in its present form does not achieve any of its objectives [29], which is in line with the results in this study. One possible explanation may be that the practitioners do not know about the ISO 9126 standard's definition of different QR, or maybe the definitions of the different QR characteristics are not important to practitioners in an industrial context.

Moreover, Berntsson Svensson et al. found that there may be a possible mismatch between the established academic interpretation of quality characteristics of ISO/IEC9126 and the industrial interpretation of it [18]. The possible identified mismatch in [18] may be an explanation of why it was difficult to classify each QR into the ISO 9126 standard's characteristics and sub-characteristics. Especially, our results indicate that there is a need to tailor the breakdown of quality attributes to the domain in question. Although

ISO 9126 Characteristics	ISO 9126 sub characteristics	Architecture	Audio	Camera	Connectivity	Display	HW architecture	IMS	Industrialization	Java	Memory	Messaging	Mobile TV	Multimedia	Network	Positioning	Power	Radio	Security	UI	Video telephony	Grand Total
Efficiency	Efficiency																	4			1	5
	Efficiency compliance														1							1
	Resource utilization	3	1	1	1				1	1	3		1		1		20	45				78
	Time behavior	8	5	8	17	1			9	1	14	1	5	30	5	1	1	130			3	239
Functionality	Accuracy		18			6										2	4	9				39
	Functionality												2					2				4
	Functionality compliance		1	2	11	5		5		19	14	2	2	12	5	2		4			2	86
	Interoperability		7		2													2				11
	Security	1			14			10	1			1				1			65			93
	Suitability	14	4		3	1				5		1			4	3	4		14			53
Maintainability	Changeability	5	2	11		3		2	1	1	2				1		1					29
	Testability	7							11				1		4		1	2				26
NA	NA	16					9								4		1	62	1			93
Portability	Adaptability	5	10			1					1				1	2	1	1	1			23
	Portability														1							1
	Portability compliance				3																	3
	Replaceability	3			1					2									1			7
Reliability	Fault tolerance	3									1											4
	Maturity																	1				1
Usability	Attractiveness		4		5															1		10
	Operability		6		1					3				4						3	3	20
	Understandability																		1			1
Grand Total		65	58	22	58	17	9	17	23	32	35	5	11	46	27	11	33	262	83	4	9	827

Fig. 6. ISO 9126 to sub-domain mapping (an empty field means no ISO characteristic found in that sub-domain), "N/A" representing requirements with no corresponding ISO 9126 characteristic.

a domain-specific method for categorizing QR requires an initial tailoring to be useful, once the categorization scheme is defined, the method can be reasonably reliable and efficient.

Looking into the types of QR that are present in the requirements specification, the results show that efficiency requirements (323 of 827) are the most specified QR type in the requirements specification, followed by functionality (286 of 827) requirements, which is not in line with the findings in de la Vara et al. [25] who found that performance requirements are the third most frequent specified QR type. The most frequent QR type in [25] is usability followed by maintainability. While de la Vara et al. [25] collected the subjects' opinions, we have studied which QR types are actually specified in a real requirements specification, which may explain the difference between the studies.

Among the sub-characteristics in Table 6 (excluding the NA category), the most frequently specified QR types are: (1) Time behavior (239 of 827), (2) Security (93 out of 827), and (3) Functionality compliance (86 out of 827). The frequency of specified time behavior and security requirements implies that these are the most important types of QR to specify, which is not in line with the results in Johansson et al. [22] who found that reliability is the most important QR. In addition, only five reliability requirements are present in the analyzed requirements specification. The difference between the studies may be explained by the collected data, i.e. we have analyzed a real requirements specification from industry and looked into the actual specified QR, while Johansson et al. [22] have asked subjects' opinions about the importance of different types of QR. Moreover, Leung found that the two most important types of QR are availability and accuracy [23], which is not in line with our results. In addition, Leung found that performance requirements (time behavior) are only considered the fifth most important quality aspect. The difference between the studies may be explained by the focus, i.e. we focused on a case company in a B2B domain using a sub-contractor requirements specification, while Leung [23] focused on intranet applications. The importance of performance requirements is in line with the findings for B2B companies in Berntsson Svensson et al. [6], which may be explained by the focus of the study. The case company's requirements specification in our study is in the B2B domain, and the study by Berntsson Svensson et al. [6] is the only one to analyze the importance of QR based on type of customers. In addition, usability requirements were not considered as important for B2B companies in [6], which is in line with our results.

6. Conclusions

In conclusion, this paper presents the results of an empirical study that examines how QR are specified in industrial practice at a case company in the mobile handset domain. Data is collected from a requirements specification written in structured natural language that contains 2178 requirements, where 827 (38%) are QR.

In relation to RQ1, how QR are distributed in a requirements specification, the findings reveal that there is a large variation across the different sub-domains. The variation is related to number of QR (ranging from 4 to 262 QR), existing QR types, and how QR are specified, e.g., number of NQQR, QQR, and use of standard references. This variation can to some part be explained by the characteristics of the different sub-domains. However, the results indicate that there is a lack of a systematic method for QR, especially since some areas have deficiencies when it comes to QR.

Although relatively many QR exists in the requirements specification, there are areas of improvement. For example, there are very few maintainability requirements in the requirements specification. As the case study company employs a platform development approach, maintainability is a key factor in keeping high quality and reducing effort of using the same platform for several products. It may be, though, that maintainability requirements are hidden in other quality requirements such as portability and functionality. Standards are commonly used in all types of requirements in the case company's requirements specification. However, the phenomena is not well understood when it comes to problems and implications of hiding requirements within standards.

The findings for RQ2, quantification of QR, show that 56% (465 out of 827) of all QR are quantified. This, and other variations across the sub-domains, suggest that methods for QR need to be able to cope with a variety of QR types. Solely focusing on, for example, quantifying QR might overlook important requirements.

The specification is written in structured natural language without explicit specification of interdependencies among requirements. As QR typically crosscut a functional decomposition, the lack of referencing structure creates obstacles. For example, FR are sometimes repeated several times for each associated QR. Another example is having a separate section for crosscutting concerns, apart from the functional structure of the specification. This causes problems with getting an overview of QR, which may lead to deficiencies in completeness and even contradicting requirements, discovered late in the process. In addition, this may lead to a maintainability problem, as requirements on one subject are spread out in the specification with little or no support for finding them in the specification. Hence, practitioners are reliant on the human factor to find the interdependencies among requirements. By improving the structure for specifying relationships among requirements, many positive effects could potentially be seen.

In relation to RQ3, existing types of QR in the requirements specification, the findings reveal that performance requirements are the most frequently specified type of QR in the requirements specification, which is not surprising considering the developed software products (small embedded system) at the case company. In addition, several of the characteristics and sub-characteristics of ISO/IEC 9126 does not exist in the requirements specification, and as many as 11% of all QR could not be classified in any of the characteristics in the ISO/IEC 9126 standard. Using standard methods such as ISO/IEC 9126 may be difficult as it fails to incorporate domain specific aspects relevant for a successful approach. Moreover, the use of ISO/IEC 9126 was more time consuming and less reliable than the tailored categories. Therefore, 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 QR. This impacts all areas of RE, starting with elicitation and analysis to specification and validation.

To complement this study, an interview study with sub-domain experts would improve the understanding of the rationale behind the specifications of requirements. The impact of standards on the requirements practice is only briefly analyzed in this study. Therefore, to further understand the impact of standards, interviews with practitioners are recommended. Moreover, further case studies are needed in other domains and on other requirements specifications to enable generalization outside the domain of this study.

Writing requirements in a structured natural language form is commonly used in industry. Despite many years of traceability research and research on dependencies among requirements, the state of practice still struggle with complex interdependencies among requirements. Therefore, it would be interesting to analyze requirements specifications focused on explicit and implicit interdependencies among requirements in general, and QR in particular. This can be performed as a document analysis study and complemented with interviews to get a comprehensive understanding of the problems. It is also important when evaluating

Table 7
Overview of sub-domains.

Sub-domain	Description
Architecture	Architecture includes requirements on the architecture as such, e.g. API requirements or componentization of software
Audio	Requirements related recording and playback of audio are found in this sub-domain, e.g. sampling rate or number of speakers
Camera	Camera includes requirements on the camera and its interfaces, e.g. resolution support and encoding of pictures
Connectivity	Requirements related to local connectivity, e.g. USB or Bluetooth™, as opposed to connections to the mobile phone network
Display	Display includes requirements on e.g. color depth, resolution or number of displays
HW architecture	Hardware requirements or mechanical requirements, e.g. component height or pin compatibility, are associated with the HW architecture sub-domain
IMS	The IP Multimedia Subsystem (IMS) is a framework for delivering Internet Protocol (IP) multimedia services to mobile devices, such as voice or chat applications
Industrialization	Industrialization includes requirements related to production of devices in factory, such as time to download software to the devices and test log requirements
Java	Java includes requirements on which Java APIs to support, such as JSR-135, and on behavior of java applications, such as memory requirements
Memory	Requirements on memories, e.g. RAM bit order or flash memories error handling, found in the Memory sub-domain
Messaging	Requirements on e.g. SMS or e-mail, are found in the Messaging sub-domain
Mobile TV	Mobile TV includes requirements on Mobile-TV enables, such as broadcast standard and recording requirements
Multimedia	Multimedia application requirements, such as encoding and decoding standard and bit rate of video, are part of the Multimedia sub-domain
Network	Requirements related to GSM or UMTS network access, both circuit-switched as well as packet-switched, are part of the Network sub-domain
Positioning	Positioning related requirements, e.g. GPS and emergency location services, are found in the Positioning sub-domain
Power	Power includes requirements on charging, batteries, etc
Radio	Everything surrounding radio access protocols, such as sensitivity and frequencies, are part of the Radio sub-domain
Security	Security includes requirements on Security, such as encryption and identification of users
UI	UI include requirements in, for example, input, such as simultaneous key presses, and output, such as mechanical feedback
Video telephony	Video telephony related requirements, such as resolution on a video call and protocols, are part of the Video Telephony sub-domain

methodologies for dependencies among requirements to keep in mind the return of investment. It is not obvious that adding explicit dependencies will be beneficial when analyzing the complete development process, since maintainability problems may occur.

Acknowledgements

This work was partly funded by VINNOVA (the Swedish Agency for Innovation Systems) within the MARS project and by the Industrial Excellence Center EASE – Embedded Applications Software Engineering, (http://ease.cs.lth.se). Furthermore, we would like to thank all of the participants and their companies who have helped in making the data collection possible for this research.

Appendix A

Table 7.

References

[1] S. Konrad, M. Gall, Requirements engineering in the development of large-scale systems, in: Proceedings of the 16th IEEE International Requirement, Engineering Conference, 2008, pp. 217–222.
[2] L. Chung, B.A. Nixon, E. Yu, J. Mylopoulos, Non-Functional Requirements in Software Engineering, Kluwer Academic Publisher, 2000.
[3] C. Ebert, Putting requirements into praxis: dealing with nonfunctional requirements, Information and Software Technology 40 (1998) 175–185.
[4] N.A. Ernst, J. Myloppulos, On the perception of software quality requirements during the project lifecycle, Lecture Notes in Computer Science 6182 (2010) 143–157.
[5] S. Barney, A. Aurum, C. Wohlin, A product management challenge: creating software product value through requirements selection, Journal of Systems Architecture 54 (2008) 576–593.
[6] R. Berntsson Svensson, T. Gorschek, B. Regnell, R. Torkar, A. Shahrokni, R. Feldt, Quality requirements in industrial practice – an extended interview study at eleven companies, IEEE Transaction on Software Engineering 38 (2012) 923–935.
[7] A. Borg, A. Yong, P. Carlshamre, K. Sandahl, The bad conscience of requirements engineering: an investigation in real-world treatment of non-functional requirements, in: Third Conference on Software Engineering Research and Practice in Sweden (SERPS 2003), Lund, Sweden.
[8] L. Chung, J.C.S. do Prado Leite, On non-functional requirements in software engineering, Lecture Notes in Computer Science 5600 (2009) 363–379.
[9] M. Glinz, On non-functional requirements, in: Proceedings of the 15th IEEE International Conference on, Requirements Engineering, RE'07, 2007, pp. 21–26.
[10] S. Jacobs, Introducing measurable quality requirements: a case study, in: Proceedings of the Fourth IEEE International Symposium on, Requirements Engineering, 1999, pp. 172–179.
[11] L. Karlsson, Å.G. Dahlstedt, B. Regnell, J. Natt och Dag, A. Persson, Requirements engineering challenges in market-driven software development – an interview study with practitioners, Information and Software Technology 49 (2007) 588–604.
[12] R. Berntsson Svensson, T. Gorschek, B. Regnell, R. Torkar, A. Shahrokni, R. Feldt, A. Aurum, Prioritization of quality requirements: state of practice in eleven companies, in: Proceedings of the 19th IEEE International Conference on Requirements Engineering, RE'11, Trento, Italy, 2011, pp. 69–78.
[13] T. Olsson, R. Berntsson Svensson, B. Regnell, Non-functional requirements metrics in practice – an empirical document analysis, Workshop on Measuring Requirements for Project and Product Success, 2007.
[14] C. Seaman, Qualitative methods in empirical studies of software engineering, IEEE Transactions on Software Engineering 25 (1999) 557–572.
[15] D. Grimshaw, G.W. Draper, Non-functional requirements analysis: deficiencies in structured methods, Information and Software Technology 43 (2001) 629–634.
[16] M. Lubars, C. Potts, C. Richter, A Review of the State of the Practice in Requirements Modelling, in: Proceedings of the 1st IEEE International Symposium on Requirements Engineering, San Diego, USA, 1993, pp. 2–14.
[17] E. Kamsties, K. Hörnmann, M. Schlich, Requirements Engineering in Small and Medium Enterprises, in: Proceedings of the International Conference on European Industrial Requirements Engineering, London, UK, 1998, pp. 84–90.
[18] R. Berntsson Svensson, T. Gorschek, B. Regnell, Quality requirements in practice: an interview study in requirements engineering for embedded systems, in: M. Glinz, P. Heymans, (Eds.), REFSQ 2009, LNCS, vol. 5512, Springer, Heidelberg, pp. 218–232.
[19] L.M. Cysneiros, J.C.S. do Prado Leite, J. de Melo Sabat Neto, A framework for integrating non-functional requirements into conceptual models, Requirements Engineering 6 (2001) 97–115.
[20] J. Doerr, D. Kerkow, T. Koenig, T. Olsson, T. Suzuki, Non-functional requirements in industry – three case studies adopting an experience-based NFR method, in: Proceedings of the 13th IEEE International Conference on Requirements Engineering, RE'05, Paris, France, 2005, pp. 373–382.
[21] M. Hassenzahl, R. Wessler, K.C. Hamborg, Exploring and understanding product qualities that users desire, in: Proceedings of the 5th Annual Conference of the Human–Computer Interaction Group of the British Computer Society, IHm-HCI 01, 2001, pp. 95-96.
[22] E. Johansson, A. Wesslen, L. Bratthall, M. Höst, The importance of quality requirements in software platform development—a survey, in: Proceedings of the 34th Annual Hawaii International Conference on System Sciences, 2001.
[23] H.K.N. Leung, Quality metrics for intranet applications, Information and Management 38 (2001) 137–152.
[24] M. Sibisi, C.C. van Waveren, A process framework for customising software quality models, in: Proceedings of the IEEE AFRICON Conference, 2007, pp. 547–554.
[25] J.L. de la Vara, K. Wnuk, R. Berntsson Svensson, J. Sánchez, B. Regnell, An empirical study on the importance of quality requirements in industry, in: Proceedings of the 23rd International Conference on Software Engineering and Knowledge, Engineering, SEKE'11, 2011, pp. 438–443.
[26] R. Berntsson Svensson, M. Höst, B. Regnell, Managing quality requirements: a systematic review, in: Proceedings of the 36th EUROMICRO Conference on Software Engineering and Advanced Applications, 2010, pp. 261–268.

[27] R. Berntsson Svensson, Y. Sprockel, B. Regnell, S. Brinkkemper, Setting quality targets for coming releases with QUPER – an industrial case study, Requirements Engineering 17 (2012) 283–298.
[28] ISO/IEC 9126–1:2001(E): Software Engineering – Product Quality – Part 1: Quality Model, 2001.
[29] H. Al-Kilidar, K. Cox, B. Kitchenham, The use and usefulness of the ISO/IEC 9126 quality standard, in: Proceedings of the International Symposium on, Empirical Software Engineering, 2005, pp. 122–128.
[30] C. Pohl, G. Böckle, F.J. van der Linden, Software Product Line Engineering: Foundations, Principles and Techniques, Springer-Verlag, New York, USA, 2005.
[31] P. Runeson, M. Höst, Guidelines for conducting and reporting case study research in software engineering, Empirical Software Engineering 14 (2009) 131–164.
[32] C. Robson, Real World Research, Blackwell, Oxford, 2002.
[33] P. Runeson, M. Höst, A. Rainer, B. Regnell, Case Study Research in Software Engineering: Guidelines and Examples, John Wiley & Sons, USA, 2012.
[34] J. Boegh, A new standard for quality requirements, IEEE Software 25 (2008) 57–63.
[35] IEEE, IEEE Recommended Practice for Software Requirements Specifications. IEEE Std. 830–1998.
[36] S. Robertson, J. Robertson, The Volere Requirements Process, Mastering the Requirements Process, Addison-Wesley, London, 1999.
[37] I. Alexander, Misuse cases help to elicit non-functional requirements, Computing & Control Engineering Journal 14 (2003) 40–45.
[38] L. Chung, S. Supakkul, Representing nFRs and fRs: a goal-oriented and use case driven approach, in: W. Dosch, R.Y. Lee, C. Wu (Eds.), SERA 2004, LNCS, vol. 3647, Springer, Heidelberg, 2006, pp. 29–41.
[39] A. Herrmann, B. Paech, MOQARE: misuse-oriented quality requirements engineering, Requirements Engineering 13 (2008) 73–86.
[40] H.-W. Jung, S.-G. Kim, C.-S. Chung, Measuring software product quality: a survey of ISO7IEC 9126, IEEE Software 21 (2004) 88–92.