Quality Requirements in Practice: An Interview Study in Requirements Engineering for Embedded Systems

Richard Berntsson Svensson[1], Tony Gorschek[2], Björn Regnell[1]

[1]Lund University, Department of Computer Science, Box 118,
221 00 Lund, Sweden
{Richard.Berntsson_Svensson, Bjorn.Regnell}@cs.lth.se
[2]Blekinge Institute of Technology, School of Engineering, PO Box 520,
372 25 Ronneby, Sweden
tony.gorschek@bth.se

Abstract. Context & motivation: In market-driven software development it is crucial, but challenging, to find the right balance among competing quality requirements (QR). **Problem:** In order to identify the unique challenges associated with the selection, trade-off, and management of quality requirements an interview study is performed. **Results:** This paper describes how QR are handled in practice. Data is collected through interviews with five product managers and five project leaders from five software companies. **Contribution:** The contribution of this study is threefold: Firstly, it includes an examination of the interdependencies among quality requirements perceived as most important by the practitioners. Secondly, it compares the perceptions and priorities of quality requirements by product management and project management respectively. Thirdly, it characterizes the selection and management of quality requirements in down-stream development activities.

Keywords: Quality requirements; Non-functional requirements; Requirements engineering; Market-driven requirements engineering; Empirical study

1 Introduction

The complexity of software systems is determined by both functionality and by quality aspects such as performance, reliability, accuracy, security, and usability [6]. These quality aspects, or non-functional requirements are subsequently called quality requirements (QR). It is commonly acknowledged that the handling and balance of QRs are an important and difficult part of the requirements engineering (RE) process [16], playing a critical role in software development [6]. However, the situation is even more complex in a market-driven development situation [1]. In market-driven development, the flow of requirements is not limited to one project, and the requirements are generated from internal (e.g., engineers) and external (e.g., customers) sources [15]. Also, to achieve high-quality in complex embedded systems, a combination of experience and knowledge from different disciplines is needed [19]. This may lead to communication difficulties and difficulties in achieving the required quality level [19]. One characteristic of QR is the specification of certain quality levels, and therefore QRs are in most cases possible to quantify [22]. This is important, not only for understanding QRs [16], but also for planning [24]. Not dealing, or ineffectively dealing with QRs may lead to more expensive software and longer-time-to-market [10], or in worst case, failures in software development [2, 11]. Studies [3, 9] have showed that QRs are the most expensive and difficult aspects to handle, and according to Chung et al., QRs are often poorly understood in comparison to less critical aspects of software development [6]. To be able to improve how QRs are handled it is important to understand their characteristics [22], how they are used and prioritized in industry, as well as the challenges of dealing with QRs. This paper presents an empirical study performed in industry to investigate these aspects as well as complement other RE surveys as few of them have focused on the specific challenges related to QRs.

Two main perspectives on QRs are studied in this paper [14]. First, the product perspective. Product managers are responsible for the overall product perspective and the selection of the overall planning of the product evolution and offering are elicited. Second, the project perspective is studied through the project leader, responsible for managing and prioritizing within the realization phases. The two perspectives are also compared, studying the alignment between project and product managers. The purpose of this study is to discover and describe how QRs are handled in practice, both from the product manager and the project leader’s perspective. In addition, the effects of not dealing, or ineffectively dealing with QR are also investigated. The paper presents the results of an empirical study that includes data collected from ten practitioners (five product managers and five project leaders) at five companies in Sweden.

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

2 Background and Related Work

There are several surveys that concern or include RE related challenges. Curtis et al. reported the first significant field survey of practices [8]. Even though the study does not have a focus on RE, challenges related to RE were identified, including communication breakdowns and conflicting requirements. Next, a study by Chatzoglou identifies problems with the RE process, the challenges presented are e.g., lack of resources and poor quality of tools and techniques in the RE process [5].

Lubars et al. published a field study on requirements modeling [21]. The presented challenges include vaguely stated requirements and difficulties with prioritization of requirements. In addition, Lubars at el. identified challenges in relation to specification of performance requirements (QR) such as the rationale is not always obvious and difficulties to associate performance requirements with parts of dataflow or control flow specifications. In addition, a field study by Kamsties et al. includes small and medium sized enterprises [17]. The identified challenges include implementation of new requirements may cause unpredictable interaction with existing requirements, requirements are not traceable, and that requirements are too vague to test. Kamsties et al. also indentified a challenge related to specification of graphical user interfaces (usability requirements, i.e. QR). Furthermore, Karlsson et al. published a study with solely focus on challenges in market-driven software development [18]. The presented challenges include communication problems between marketing and development, and requirements prioritization. Karlsson et al. also indentified challenges in relation to QRs. One challenge is related to QRs interdependencies, which was identified as a major problem. Quality requirements can influence a large part of the functionality or other QRs. This is not only related to find the existing interdependencies, but also how the requirements affect each other, and how to deal with it. In addition, problems with considering quality requirements in release planning were identified.

Several studies [4, 6, 7] have looked at requirements interdependencies; for example, Carlshamre et al. identified six different interdependency types in industry [4]. Research related to classification and measurement of QRs are also introduced in literature [16, 22]. Olsson et al. conclude that for a method to be successful, it is important that it is flexible enough to handle the diverse nature of QR [22].

The focus of the above mentioned studies have not been on the critical aspects of handling QRs in the RE process, or to increase the awareness and understanding of QR specifically.

3 Research Method

In an attempt to understand the industrial perspective on QR and handling of non-functional aspects, this paper investigates the nature of QRs and some associated challenges. The study was carried out using a qualitative research approach [25]. Qualitative research aims to investigate and understand phenomena within its real life context. 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. The purpose of this study is to gain in-depth understanding of QRs within market-driven software companies. The following research questions (see Table 1) provided a focus for our empirical investigation.

It is important to understand an organizations alignment in terms of QR, otherwise there may be a mismatch between Product Management (PM) and Project Leader (PL). The PL may down prioritize quality aspects that are considered important by the PM and vice verca. In addition, interdependencies are important to understand since QR may influence a large part of the system [18]. Kamsties et al. found that requirements are too vague to test [17], therefore, it is important to investigate if QRs are quantified in industry. Also, dismissal of QR from projects may have an impact on the predicted return of investment, as well as the cost for the customers. Finally, QRs are a difficult part of the RE process [16], however; not all challenges in relation to QR may be of major concern for industry. Therefore, it is important to understand what challenges are critical and which ones are adequately handled today.

Table 1. Research questions

Research Questions
RQ1: Is there any difference between the view of what quality requirements are the most important between product manager and project leader?
RQ2: What interdependencies between QR are present in the companies?
RQ2.1: What types of interdependencies are deemed most important by practitioners, and is there any difference between the view of product managers and project leaders in this regard?
RQ2.2: To what extent are interdependencies elicited, analyzed and documented in industrial practice?
RQ3: Are QR specified in a measurable manner?
RQ4: To what extent are QR dismissed from projects after project initiation?
RQ4.1: If QR are dismissed, is any consequence analysis performed?
RQ5: What QR challenges are articulated as critical by the practitioners themselves?
RQ6: What QR aspects do the companies feel confident as being adequately handled today?

QR: Quality Requirements

3.1 Research Design and Data Collection

The study uses semi-structured interviews enabling exploratory discussion between the researcher and the interviewee. The study was conducted in two stages: first the data from each company was collected and analyzed. Secondly, the combined data from all participating companies was collected and analyzed. The criteria for selecting companies were based on our corporate contacts within industry. Five market-driven software companies participate. From each company, one product manager (PM) and one project leader (PL) from the same project were interviewed, resulting in ten data points. The study consists of three phases: planning, data collection, and analysis.

Planning: The first phase of the study involved a brainstorming and planning meetings to design the study and to identify different areas of interests. A combination of maximum variation sampling and convenience sampling was used to select companies within our industrial collaboration network [23]. The included companies vary in respect to size, type of product, and application domain, a rudimentary characterization can be see in Table 2. The interview instrument was designed with respect to the different areas of interest and inspiration from [18]. To test the interview instrument[1], two pilot interviews were conducted prior to the industry study.

Data collection: The study used a semi-structured interview strategy [25]. All interviews were attended by one interviewee and one interviewer. First, the purpose of the study was presented and then questions about the different areas of interests in

[1] http://serg.cs.lth.se/research/packages

relation to QRs were discussed in detail. All interviews varied between 40 and 90 minutes.

Analysis: The content analysis [25] involved creating categories where interesting parts from the interviews were added and discussed. The first two authors examined the categories from different perspectives and searched for explicitly stated or concealed pros and cons in relation to how QRs are handled in industry. The results from the analysis are found in section 4.

Table 2. Company characteristics

	Aplha	Beta	Gamma	Delta	Epsilon
Number of employees	~100	~3000	>5000	325	65
Product in focus	Control systems	Mobile platforms	Mobile handsets	Mobile applications	Autmomated guided vehicles and software intensive systems
Customer	Diary industry and oil rigs	Mobile handset vendors	Network operators and the global market	Network operators and mobile handset vendors	Partners and resellers
Typical project cycle	18 months	48 months	24-36 months	Differs	9 months
Number of requirements	Thousands, ~10% QR	~7000, ~10% QR	>20000, QR unknown	~100 features, ~10% QR	Differs

3.2 Validity

In this section, threats to validity are discussed. We consider the four perspectives of validity and threats presented in Wohlin et al. [26].

Construct validity: The construct validity is concerned with the relation between theories behind the research and the observations. The variables in our research are measured through interviews, including open-ended aspects where the participants are asked to express their own opinions. Mono-operation bias [26] was avoided by collecting data from a wide range of sources on the topic of the study. To avoid evaluation apprehension [26], complete anonymity from other participants, the companies, and researchers was guaranteed. Another validity threat lies in the question that asked interviewees to rank and include additional factors if the list provided to them was inadequate. Interviewees may have thought that it was easier to rank the provided factors than propose new factors. This means that important interdependency types may be missing.

Conclusion validity: Threats to conclusion validity arise from the ability to draw accurate conclusions. The interviews were conducted at different companies and each interview was done in one work session. Thus, answers were not influenced by internal discussions. To obtain highly reliable measures and to avoid poor question wording and poor layout, several pilot studies were conducted.

Internal validity: This threat is related to issues that may affect the causal relationship between treatment and outcome. Threats to internal validity include instrumentation, maturation and selection threats. In our study, the research instrument was developed with close reference to literature relating to non-functional requirements, and influenced by a previously administrated and validated research instrument [18], which mitigates the instrumentation threat. In addition, maturation threats are handled by reducing the duration of interview sessions by collecting background information before the interview, and by keeping the interview session to 90 minutes.

External validity: This threat is concerned with the ability to generalize the findings beyond the actual study. Qualitative studies rarely attempt to generalize beyond the actual setting since it is more concerned with explaining and understanding the phenomena. However, understanding the phenomena may help in understanding other cases. The fact that most of the identified challenges are acknowledged by more than one company increases the possibility to generalize the results beyond this study. To avoid the interaction of selection and treatment, interviewees were selected according to their roles within the company, and companies were selected from different geographical locations.

4 Results and Analysis

This section presents the results discovered during the analysis of the interviews. The five following sub-sections present and discuss one research question each, corresponding to the research questions in Table 1.

4.1 Important Quality Aspects (RQ1)

In analyzing Research Question 1, this section examines the most important quality aspects, as illustrated in Figure 1. Based on the comparison of ISO9126 and McCall quality factors [20], we identified 23 different types of QRs. We asked the interviewees to rank the top five most important aspects for their products based on their expertise. Looking at Figure 1, the quality aspect ranked first received five points, the one ranked second got four points and so on, and the one ranked fifth got one point. In total we see that interviewees agreed that usability (which got a total of 26 points) and performance (23 points) requirements are the two most important types of QRs followed by compliance (13 points), flexibility (13 points), and stability (11 points).

One reason for the prioritization of usability, as explained by several interviewees, is that *"if the product is not usable we will not sell any products"*. One interviewee expanded the view by stating that it does not matter if you have the latest and coolest functionality, if the system is not easy to use, the customer will look at the competitors for an easy to use system. The reason why compliance was ranked as the third most important quality aspect is interesting. Several interviewees explained that compliance is important because *"we must be compliant with the requirement document"*. This interpretation of compliance differs from the one formulated by ISO9126 which states to adhere to standards, regulations and laws. This leads to a possible mismatch between the established academic interpretation of compliance and the industrial interpretation of it.

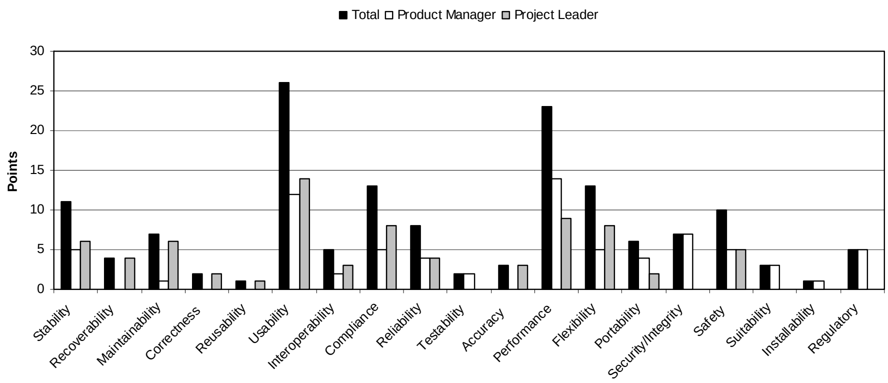

Figure 1. Importance of quality aspects

Apart from the agreement that usability and performance were the two most important types of QRs, PM and PL had different priorities. PMs ranked performance (14 points) as the most important quality aspect, followed by usability (12 points) and security/integrity (7points). PLs ranked usability first (14 points), followed by performance (9 points), and compliance and flexibility (8 points). PMs uniquely identified security/integrity, testability, suitability, and installability as the most important quality aspects. On the other hand, PLs uniquely identified the following quality aspects, recoverability, reusability, correctness, and accuracy. The differences in priority between PM and PL may not be a surprise as some mismatch can be expected. The two have different roles and perspectives, but it might be an important insight nevertheless. For example, not a single PL ranked security/integrity among the top five even if security/integrity was considered the third most important by the PMs. Worth observing is that the aspects of fault tolerance, conformance, replaceability, and analyzability was not in any of the PMs or PLs top five.

4.2 Interdependencies (RQ2)

Six different interdependency types are characterized [4]: **(1)** R_1 *AND* R_2: R_1 requires R_2 to function, and R_2 requires R_1 to function, **(2)** R_1 *REQUIRES* R_2: R_1 requires R_2 to function, but not vice versa, **(3)** R_1 *TEMPORAL* R_2: Either R_1 has to be implemented before R_2 or vice versa, **(4)** R_1 *CVALUE* R_2: R_1 affects the value of R_2 for a customer, **(5)** R_1 *ICOST* R_2: R_1 affects the cost of implementing R_2, and **(6)** R_1 *OR* R_2: Only one of $\{R_1, R_2\}$ needs to be implemented.

Although the interviewees had the option of adding new types of interdependencies, no new types were discovered during the interviews. All of the six presented interdependency types were used by the interviewees to characterize perceived interdependencies, both between different QRs, and interdependencies between QRs and functional requirements (FR), as illustrated in Table 3.

In general, the most common interdependency types identified between QRs were: *OR*, *REQUIRES*, and *ICOST*, while the least frequent one identified was

TEMPORAL. When the results from PM and PL were examined separately, the findings show a difference of opinion. PMs viewed *OR* and *ICOST* as the most common types, while PLs viewed *REQUIRES* as the most common one.

When examining the most frequent identified interdependency types between QR and FR, four of the six types were considered equally common, while the remaining two (*AND*, *OR*) types were considered least important. However, when examining the results from the PM and PL separately, the findings show an interesting difference. While PM considered *TEMPORAL* as the most common interdependency, PL viewed *TEMPORAL* as least frequent. On the other hand, PL identified *OR* as one of four (the other three are: *REQUIRES*, *ICOST*, and *CVALUE*) equally common interdependency types, but *OR* was viewed as least frequent by PM.

Table 3. Existing interdependency types divided by role

	Role	QR to QR	QR to FR
Alpha	*PM*	REQUIRES, **CVALUE**, ICOST	AND, REQUIRES, TEMPORAL, CVALUE, **ICOST**
	PL	NONE	NONE
Beta	*PM*	OR, AND, REQUIRES, TEMPORAL, CVALUE, **ICOST**	OR, AND, REQUIRES, TEMPORAL, CVALUE, **ICOST**
	PL	OR, **REQUIRES**, TEMPORAL, CVALUE, ICOST	OR, **REQUIRES**, TEMPORAL, CVALUE, ICOST
Gamma	*PM*	OR, AND, **REQUIRES**, CVALUE, ICOST	AND, **REQUIRES**, TEMPORAL, CVALUE, ICOST
	PL	OR, AND, REQUIRES, **CVALUE**, ICOST	AND, REQUIRES, **CVALUE**, ICOST
Delta	*PM*	OR, **ICOST**	OR, REQUIRES, TEMPORAL, CVALUE, **ICOST**
	PL	OR, **REQUIRES**, CVALUE, ICOST	OR, AND, **REQUIRES**, TEMPORAL, CVALUE, ICOST
Epsilon	*PM*	**OR**	**TEMPORAL**
	PL	**REQUIRES**	**OR**

In a study by Carlshamre et al., three of five case companies viewed value related (*ICOST* or *CVALUE*) interdependency types as the most common [4]. In the remaining two cases, functionality related (i.e., *AND* or *REQUIRES*) types were most common. Our results show a mix of value and functionality types as the most common ones (with the exception of Company Epsilon). The difference between the studies may be explained by the focus, i.e.we focused solely on interdependencies related to QR, while in Carlshamre et al. the focus was on requirements in general [4]. In [6, 7], a softgoal interdependency graph (SIG) is used to show interdependencies between QRs. The interdependency types used in the SIG are limited to *AND*, and *OR*, which is not inline with the findings in our study, as we found that six different interdependency types were present in the companies. Furthermore, the two types *AND*, and *OR* were only identified as present by 25% of the interviewees.

RQ2.1: What types of interdependencies are deemed most important by practitioners, and is there any difference between the view of product managers and project leaders in this regard? According to the interviewees in total, the most important interdependency type to identify between QRs was *REQUIRES*, however, the PM and PL roles were not in agreement. PM considered *ICOST* as the most important, while *REQUIRES* was prioritized by the PL. Interestingly, in identifying the most important interdependency type between QR and FR, the total result was identical to interdependency types between QRs. On closer examination the result between PM and PL vary in relation to interdependencies between QR and FR. PM prioritized *ICOST*, but also uniquely identified *TEMPORAL* and *ICOST*. The PL prioritized *REQUIRES*, but also uniquely identified *OR* and *CVALUE*.

It is not surprising that PM and PL have different views on interdependency priority. According to Carlshamre et al., value related interdependencies are subjective; it may be difficult to state whether the cost exceeds the value for the customer, therefore, these types of decisions should be made by product committees [4]. This is inline with the results in this study; PM considers *ICOST* as the most important type, while PL *REQUIRES*. One PL explained that *REQUIRES* is the most important interdependency type because *"functionality first, then the quality aspect of the functionality is relevant"*. Surprisingly, both between QRs, and between QR and FR, *REQUIRES* is considered the most important to identify looking at the summation of all interviewees. This result is not inline with Carlshamre et al., which found that *ICOST* and *CVALUE* were the most important types of interdependencies in market-driven developing companies, while *REQUIRES* was considered the most important in bespoke developing companies [4]. One PM explained that *REQUIRES* is considered the most important interdependency to identify because *"this is the easiest type to miss, and therefore the most important to identify"*.

RQ2.2: To what extent are interdependencies elicited, analyzed and documented in industrial practice? The results show that in three of the five companies (Gamma, Delta, and Epsilon) both PM and PL confirmed that no elicitation, analysis, or documentation of interdependencies involving QR was conducted at all. In Company Alpha, the PM stated that all dependency activities were conducted, while the PL from the same company indicated that none of them were performed. In only one company (Beta) both PM and PL stated that activities to elicit, analyze and document interdependencies was performed. This result is inline with results from Karlsson et al., which found that interdependencies between requirements in market-driven software development are a major problem [18]. The problem includes identification, how the requirements affect each other, and how to deal with them. The results are relevant since interdependencies between QR's are at the hart of managing explicit trade-offs among solution alternatives [10]. In addition, Cleland-Huang et al. states that failing to trace QRs expose a company to huge risks when a change is introduced [7]. Furthermore, Kamsties et al. found that new requirements may cause unpredictable interaction with existing requirements, which indicates the importance of finding the interdependencies between requirements [17].

There can be several potential explanations of why interdependencies between QRs are not actively looked for. Quality requirements tend to have a global impact on the entire system, therefore, QRs are difficult to trace and because of the extensive network of interdependencies and trade-offs that exists between them responsibilities for their realization is often vague [6, 10]. Other explanations were discovered during the interviews. Some interviewees stated that they have little focus on QR, while others stated that QRs are assumed and therefore interdependencies are not actively looked for. In addition, one interviewee confirmed that their focus is on functional requirements and not QRs. Others stated that dependencies are handled during other parts of the development process, for example, during the design, architecture, and implementation. However, they have more focus on functional requirements because functional requirements are easier to discover than QRs.

One possible implication with this is that quality aspects such as usability and performance are not considered at the early stages of product and project planning. This can be an acceptable alternative, given that the companies consider quality aspects important only in the solution domain, and not from a product offering or business perspective. This is however contradicted by the results obtained during the prioritization of quality aspects (see Section 4.1), where the practitioners stated that several (or which usability was premiered) quality aspects were crucial for being able to sell the product at all.

4.3 Quantification of Quality Requirements (RQ3)

In analyzing research question 3, this section examines how often QR are specified in a measurable manner, as illustrated in Table 4.

Table 4. Quantification of quality requirements

Role	**Alpha**	**Beta**	**Gamma**	**Delta**	**Epsilon**
PM	Always	Never	Always	Sometimes	Always
PL	Sometimes	Sometimes	Always	Sometimes	Sometimes

Interestingly, four of the PLs claimed that QRs were quantified *sometimes*, while in three of these cases the PMs view differed, stating *always* or *never*. In two out of five (Gamma and Beta) companies agreement between PM and PL could be observed. The disagreement may be an indication of communication problems between the PM and PL. Communication problems were also identified as a challenge in market-driven RE by several studies [12, 13, 18]. In a study by Olsson et al., about half of the QRs were found to be quantified which seems to confirm the findings [22]. However, one interesting observation that can not be directly confirmed is the level of disagreement between PM and PL. It should be noted that each PM and PL pair worked for the same company, and moreover with the same project.

4.4 Dismissal of Quality Requirements (RQ4)

We asked the interviewees how often QR that were actually specified and selected for inclusion in a project were subsequently dismissed from project during development (see Table 5). The total average mean value of dismissed QR is 22.5%, meaning almost every fourth QR that has been included in a project is dismissed at some stage. When comparing PM and PL, the least (in the best situation) amount of dismissed QR is slightly higher for PM (5%) than for PL (3%). In worst case (*Most* in Table 5); the mean value of dismissed QRs is 55% according to the PMs, while PLs believe that 45% are dismissed.

Table 5. Dismiss rate of quality requirements

	Role	Dismissal rate			Consequence Analysis	Reason for dismiss rate
		Least	*Average*	*Most*		
Alpha	*PM*	10%	15%	20%	If customer is affected	Poor cost estimations
	PL	0%	50%	90%	No	Testing QR very late
Beta	*PM*	10%	20%	90%	If customer is affected	Lack of resources
	PL	1%	5%	20%	Yes	Lack of resources and poor cost estimations
Gamma	*PM*	NA	NA	NA	Check with stakeholders	Poor cost estimations and lack of resources
	PL	NA	NA	NA	No	Lack of resources and lower priority than FR
Delta	*PM*	0%	5%	10%	If customer is affected	Issues we cannot affect, e.g. network capacity
	PL	0%	10%	20%	No	Issues we cannot affect, e.g. network capacity
Epsilon	*PM*	0%	50%	100%	If customer is affected	Poor cost estimations and lower priority than FR
	PL	10%	25%	50%	No	Lower priority than FR

NA: *Not available*

According to the interviewees, there are two trends of which types of QRs that are more representative of the ones being dismissed. Firstly, QRs that are not visible for the end customer, such as maintainability and testability are more often dismissed than other QRs. Secondly, performance requirements are more often dismissed due to the difficulties of estimating them. One inherent contradiction can be seen in these two trends. For example, if the performance of a system is inadequate, the inadequacy of this quality aspect can be noticed by the customer through a slow system/product. No further elaborations were given on this contradiction.

The results reveal three main reasons for the dismissal of QRs: (1) poor cost estimations, (2) lack of resources, and (3) that QRs have lower priority than functional requirements (FR). Poor cost estimations is related to the difficulties to estimate the cost of QRs that have a global impact on the system. The difficulties of estimating the cost of QRs are related to lack of knowledge and understanding of how to manage QRs in practice. Several interviewees frequently described that QRs have lower priority, and that they do not spend much time on managing QRs. Some of the interviewees explained that QRs are seen as base requirements and therefore not considered. However, this focus has implications on the system, as explained by one

PM, *"in most situations, QRs are down prioritized by FR due to lack of knowledge of how important a system's quality is. By lowering the quality level, the value of the system decreases"*.

RQ4.1: If QR are dismissed, is any consequence analysis performed? According to the PMs, a consequence analysis is only conducted if the customers are affected. The consequence analysis may include new prioritization of all requirements and new cost estimations, as explained by one interviewee that *"if we have promised a certain quality, then we have to increase the cost for this project and accept a lower return of investment"*. Another consequence, as explained by a PM, is to *"first ask the customer if this is OK. If not, we talk to the developers to find out the reason why this cannot be done. Finally, we decide if we have to add or remove other requirements"*. Surprisingly, none of the PLs shared the view of the PMs. All PL claimed that nothing happens when QRs were dismissed from the projects. One explanation, which was qualified by one PL, is that *"we do not have time to re-analyze the consequence of QRs, other things are more important"*. Another explanation according to another PL is that *"we can deliver on time if QRs are dismissed"*.

A central issue here seems to be the difficulty to properly quantify as well as estimate the cost of implementing a QR, but more importantly the value of a QR. This might indicate a lack of estimation models/techniques for QR. The complexity is of course that a QR often implies a quality aspect of a system/product. Such a quality aspect is often not realized as a feature, but rather implies that all development be in line and adhering to the quality aspect. For example, performance is not dictated by one thing, but often by how the system is realized overall, including architectural considerations impacting the whole.

4.5 Quality Requirement Challenges (RQ5 and RQ6)

In analyzing research questions 5 and 6, this section examines what QR challenges and what QR aspects the practitioners identified. Figure 2 shows the two perspectives.

Three companies (Beta, Gamma, and Epsilon) stated that they are very good in terms of testing QRs (QRs that are well specified and quantified). This was confirmed by one interviewee: QRs that are quantified are easy to test. Another interviewee explained that their company has a well established test organization and good methods for testing QRs, both in lab and field environments. However, one of the identified challenges is difficulties in achieving testable QRs, i.e. making QR well specified and quantified. This is not a surprising result and is confirmed in previous studies [17, 18, 21]. Apart from the agreement of testing QRs, each company identified issues in relation to QRs that are adequately handled today. One surprising finding was that one company (Delta) stated that they are good at rejection of QRs. The product manager explained that *"we are very good in negotiation of QRs, which is to make sure that QRs are not part of the contract."*

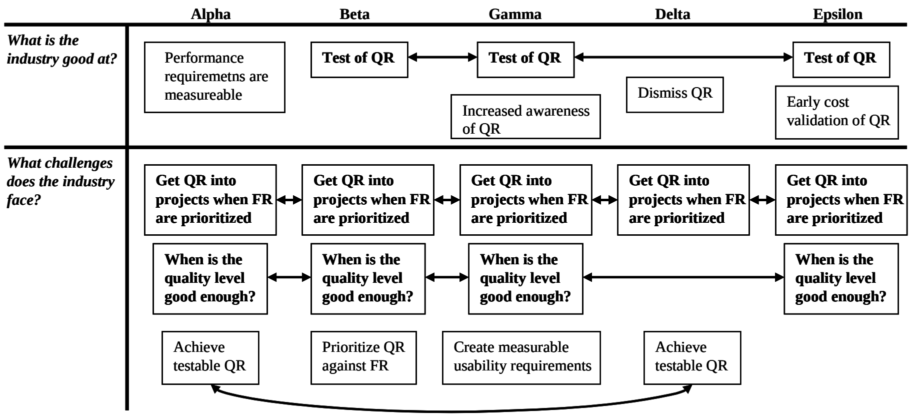

Figure 2. Challenges and non-challenges in the companies

The result reveals two major challenges that are faced by the companies, (1) how to get QRs into the projects, and (2) when is the quality level good enough? All companies faced the same problem of getting QRs into the project. The challenge is that QRs have to contend with FR, where FR often emerges as the victor. Problems with considering QRs were also found by Karlsson et al. [18]. The reason is that having an extra function is considered more valuable than to improve the quality of the system. However, this focus may backfire as the customers want a certain quality level of the systems that are bought. One interviewee confirmed that *"we have been very technology focused, we did not care about QRs, but now it has backfired and we have to put a lot of focus on the QRs."* In addition, QRs are considered as obvious, or even as base requirements and therefore not quantified or specified. The second main challenge is to decide when a certain quality level is good enough, when are you finished with a QR? The interviewees expressed their concern of how to decide when the quality is good enough. Should the performance be two seconds, 1.5 seconds, or even one second, who can decide that? One interviewee said, who can decide if 1 or 5 Mbits are most appropriate?

Two companies (Alpha and Delta) identified achieving testable QRs, and one company (Gamma) viewed creating measurable usability requirements as challenges. One reason for identifying these challenges may be related to the quantification of QRs (Section 4.3), which shows that 60% of the interviewees stated that QRs are *never*, or *sometimes* specified in a measurable manner. Another identified challenge (Company Beta) is prioritization of QRs. Prioritization of QRs faces other challenges than prioritization of FR, which were further explained by one interviewee, performance and usability requirements are different in nature and very difficult to compare. How do we prioritize performance requirements of two seconds against the subjective appraisal of usability requirements, was asked by one interviewee.

5 Conclusions

In conclusion, this article presents the results of an empirical study that examines Quality Requirements (QR) in practice in five software companies. Data is collected from five product managers and five project leaders at the companies. To the best of our knowledge, there are no other field studies that examine QR in practice.

The findings reveal that usability and performance requirements are deemed the two most important types of QRs by the interviewed practitioners. In addition, we found that the companies do not actively look for interdependencies among QRs. That is, no QR-specific elicitation, documentation, or analysis is conducted. The findings highlight three important challenges: (1) how to get QRs into the projects when functional requirements are prioritized, (2) how to know when the quality level is good enough, and (3) how to achieve testable QRs. Our results indicate that QRs are often not quantified, thus difficult to test. However, the interviewees consider that the companies are good in terms of testing the quantified QRs.

There seems to be a bespoke development mindset where the immediate project gets a higher priority than the long-term evolution of the product. This is confirmed by the implicit management of QR, and the dismissal off-hand of QR with little or no consequence analysis. This contradicts the interviewees' initial view (RQ1) where quality aspects were labeled as critical, but looking at practice, the project-oriented perspective and the urge of offering more functionality in the immediate release dominates.

The interviewees expressed that the limited focus on QRs can have long-term consequences as well; increased maintenance costs and degradation in usability with feature growth are but a few examples. However, the main problem is that QRs are not taken into consideration during product planning (pre-project) and thus not included as hard requirements in the projects. This implies that no explicit trade-off can be made, making the realization of QRs a reactive effort. Product management can thus not plan and rely on quality aspects to be competitive advantages, but only possible problems.

References

1. Aurum, A., Wohlin, C. (eds.): Engineering and Managing Software Requirements. Springer, New York (2005)
2. Breitman, K.K., Leite, J.C.S.P., Finkelstein, A.: The World's Stage: A Survey on Requirements Engineering Using a Real-Life Case Study. Journal of the Brazilian Computer Scociety 6, 13-38 (1999)
3. Brooks Jr., F.P.: No Silver Bullet: Essences and Accidents of Software Engineering. Computer 4, 10-19 (1987)
4. Carlshamre, P., Sandahl, K., Lindvall, M., Regnell, B., Natt och Dag, J.: An Industrial Survey of Requirements Interdependencies in Software Product Release Planning. In: Proc. 5th IEEE Int. Sypm. on Requirements Engineering. Los Alamitos USA, pp. 84-91 (2000)
5. Chatzoglou, P.D.: Factors Affecting Completion of the Requirements Capture Stage of Projects with Different Characteristics. Information and Software Technology 39, 627-640 (1997)

6. Chung, L., Nixon, B.A., Yu, E., Mylopoulos, J.: Non-Functional Requirements in Software Engineering. Kluwer Academic Publishers (2000)
7. Cleland-Huang, J., Settimi, R., BenKhadra, O.: Goal-Centric Traceability for Managing Non-Functional Requirements. In: Proc. 27th Int. Conf. on Software Engineering. Saint Louis USA, pp. 362-371 (2005)
8. Curtis, B., Krasner, H., Iscoe, N.: A Field Study of the Software Design Process for Large Systems. Communications of the ACM 31, 1268-1287 (1988)
9. Cysneiros, L.M., Leite, J.C.S.P.: Integrating Non-Functional Requirements into Data Model. In: Proc. 4th IEEE Int. Symp. on Requirements Engineering. Limerick Ireland, pp. 162-171 (1999)
10. Cysneiros, L.M., Leite, J.C.S.P.: Nonfunctional Requirements: From Elicitation to Conceptual Models. IEEE Transactions on Software Engineering 30, 328-349 (2004)
11. Finkelstein, A., Dowell, J.: A Comedy of Errors: The London Ambulance Service Case Study. In: Proc. 8th Int. Workshop on Software Specification and Design. Los Alamitos USA, pp. 2-4 (1996)
12. Fricker, S., Gorschek, T., Glintz, M.: Goal-Oriented Requirements Communication in New Product Development. 2nd Int. Workshop on Software Product Management. Barcelona Spain (2008)
13. Fricker, S., Gorschek, T., Myllyperkiö, P.: Handshaking between Software Projects and Stakeholders Using Implementation Proposals. In: Sawyer, P., Paech, B., Heymans, P. (eds.) REFSQ 2007. LNCS, Vol. 4542, pp. 144-159. Springer, Heidelberg (2007)
14. Gorschek, T., Davis, A.: Requirements Engineering: In Search of the Dependent Variables. Information and Software Technology 50, 67-75 (2008)
15. Gorschek, T., Wohlin, C.: Requirements Abstraction Model. Requirements Engineering Journal 11, 79-101 (2006)
16. Jacobs, S.: Introducing Measurable Quality Requirements: A Case Study. In: Proc. 4th IEEE Int. Symp. on Requirements Engineering. Limerick Ireland, pp. 172-179 (1999)
17. Kamsties, E., Hörnmann, K., Schlich, M.: Requirements Engineering in Small and Medium Enterprises. In: Proc. Conf. on European Industrial Requirements Engineering. London UK, pp. 84-90 (1998)
18. Karlsson, L., Dahlstedt, Å.G., Regnell, B., Natt och Dag, J., Persson, A.: Requirements engineering challenges in market-driven software development – An interview study with practitioners. Information and Software Technology 49, 588-604 (2007)
19. Kusters, R.J., Solingen, R.V., Trienekens, J.J.M.: Identifying Embedded Software Quality: Two Approaches. Quality and Reliability Engineering International 15, 485-492 (1999)
20. Lauesen, S.: Software Requirements – Styles and Techniques. Addison-Wesley, Great Britain (2002)
21. Lubars, M., Potts, C., Richter, C.: A Review of the State of the Practice in Requirements Modelling. In: Proc. 1st IEEE Int. Symp. on Requirements Engineering. San Diego USA, pp. 2-14 (1993)
22. Olsson, T., Berntsson Svensson, R., Regnell, B.: Non-functional requirements metrics in practice – an empirical document analysis. Workshop on Measuring Requirements for Project and Product Success. Palma de Mallorca Spain (2007)
23. Patton, M.Q.: Qualitative Research and Evaluation Methods. Sage Publications, USA (2002)
24. 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: Sawyer, P., Paech, B., Heymans, P. (eds.) REFSQ 2007. LNCS, Vol. 4542, pp. 277-191. Springer, Heidelberg (2007)
25. Robson, C.: Real World Research. Blackwell, Oxford (2002)
26. Wohlin, C., Runeson, P., Höst, M., Ohlson, C., Regnell, B., Wesslén, A.: Experimentation in Software Engineering: An Introduction. Kluwer Academic, Boston (2000)