Prioritization of Quality Requirements: State of Practice in Eleven Companies

Richard Berntsson Svensson[1], Tony Gorschek[2], Björn Regnell[1], Richard Torkar[2], Ali Shahrokni[3], Robert Feldt[3],
Aybuke Aurum[4]

[1]Department of Computer Science
Lund University
Lund, Sweden
{richard.berntsson_svensson, bjorn.regnell}@cs.lth.se

[2]School of Computing
Blekinge Institute of Technology
Karlskrona, Sweden
{tony.gorschek, richard.torkar}@bth.se

[3]Software Engineering and Technology
Chalmers University of Technology
Göteborg, Sweden
{nimli, robert.feldt}@chalmers.se

[4]School of Information Systems, Technology and
Management
University of New South Wales
Sydney, Australia
aybuke@unsw.edu.au

Abstract—**Requirements prioritization is recognized as an important but challenging activity in software product development. For a product to be successful, it is crucial to find the right balance among competing quality requirements. Although literature offers many methods for requirements prioritization, the research on prioritization of quality requirements is limited. This study identifies how quality requirements are prioritized in practice at 11 successful companies developing software intensive systems. We found that ad-hoc prioritization and priority grouping of requirements are the dominant methods for prioritizing quality requirements. The results also show that it is common to use customer input as criteria for prioritization but absence of any criteria was also common. The results suggests that quality requirements by default have a lower priority than functional requirements, and that they only get attention in the prioritizing process if decision-makers are dedicated to invest specific time and resources on QR prioritization. The results of this study may help future research on quality requirements to focus investigations on industry-relevant issues.**

Keywords-Quality Requirements; Empirical Study; Requirements Prioritization; Product Management; Project Management; Non-functional requirements;

I. INTRODUCTION

Requirements engineering is a decision-centric process [1], and decision support plays an important role in enabling the delivery of value to stakeholders [22]. Hence, decision support is crucial in achieving value to stakeholders. This is further aggravated in market–driven incremental development, where the situation is even more complex [2], due to that 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 [11].

To deliver business value, a key issue is to decide what to develop; therefore, it is important to make trade-offs between different requirements and stakeholders [29]. Requirements prioritization is an important part in requirements negotiation and release planning [22][29].

For a product to be successful, it is not enough to fulfill the functional requirements (FR), it is crucial, and challenging, to find the right balance among competing quality requirements (QR). Although literature offers many methods for requirements prioritization, the research on prioritization of QR is limited [6][12].

This paper presents the results of an empirical study that includes data collected through in-depth interviews with 22 practitioners from 11 different companies. The study focuses on prioritization of QR in industry. This exploratory study can be seen as a study of state-of-practice in industry, but also an investigation of how state-of-the-art in research, in terms of methods and techniques, has penetrated industrial practice.

The study incorporates two main perspectives with regards to prioritization of QR [8], through the study of two roles central to decision making in relation to QR. First, the product perspective is studied through the role of the Product Manager (PM), responsible for the overall product perspective and the selection of the overall planning of the product evolution and proposition offering. Second, the project perspective is studied through the role of the Project Leader (PL), responsible for managing and prioritizing requirements within the realization phases.

The reminder of this paper is organized as follows. In Section II, related work is presented. The research methodology is described in Section III, and Section IV presents the results and relates the findings to previous studies. Section V gives a summary of the main conclusions.

II. RELATED WORK

The quality of a software product is often considered as the ability of the product to satisfy customer and user needs. To increase the chance of a successful product, it is important to find, select, and plan the right releases with suitable requirements [4]. If the "wrong" requirements are selected and implemented in the product, users may resist buying the product [4]. Decision-makers often face the challenge of having more requirements than are possible to implement given different constraints, such as time, cost, and other scarce resources. Therefore, it is crucial to distinguish the important requirements from the less important ones to maximize the overall business value [30]. To find the most

important requirements that add most value to business, several requirements prioritization approaches are introduced in the literature. A selection of prioritization techniques from literature is summarized in the following subsection.

A. Requirements Prioritization Techniques

Numerical Assignment (Grouping) is, according to [4], [21], the most traditional and common prioritization technique. Numerical assignment is based on grouping requirements into different categories, where three groups are common in practice [32]. According to [33], using categories like high, medium, and low may confuse the stakeholders since different stakeholders may have different views of what, e.g. high and medium means.

The pair-wise comparison technique suggested by Karlsson [15] is based on the Analytical Hierarchy Process (AHP) [31]. In this technique, pairs of requirements are compared according to their importance. The comparisons provide an understanding of each requirements share of the total value.

Cost-value approach [16] is a prioritization technique based on AHP. The cost-value approach uses a two-dimension graph that displays the requirements value against its cost. AHP is used from a customer and user perspective to assess the value of each requirement, followed by an assessment of the requirements cost from an implementation perspective. The next step is to plot these into a cost-value diagram, which is used to analyze and discuss the requirements.

Cumulative voting ($100-Dollar Test) is a straightforward prioritization technique where stakeholders are given a fictitious $100 to distribute on requirements. When the money has been distributed, the requirements are ranked so that the highest total reflects the most important requirement, the next highest is the next-most important, etc. on a ration scale.

Ranking is a technique based on an ordinal scale where requirements cannot be tied in ranking, which means the most important requirement is ranked first, and the least important ranked last, on an ordinal scale. It is not possible to see the relative priority difference among the requirements. There are a variety of ways to rank the requirements, e.g. using bubble sort or binary search tree [17].

B. Empirical Studies of Requirements Prioritizaion

So far, to the best of our knowledge, we know of no empirical studies of with specific focus on how QR prioritization is conducted in practice [6]. In this section, we therefore describe a selection of empirical studies based on prioritization of requirements in general.

Karlsson conducted an empirical comparison of the pair-wise comparison technique and numeral assignment technique [15]. Five participants applied the techniques on 14 requirements. Karlsson found that the relative prioritization by pair-wise comparison and that the judging

of a requirements relative importance to the other tends to be more accurate than assigning absolute numbers.

Karlsson et al. conducted a self-experiment to compare six prioritization techniques [17]. All three authors prioritized 13 QR using each technique. Karlsson et al. concluded that the AHP was the most promising technique due to providing the most trust worthy results, and it includes a consistency check [17]. However, scalability was identified as a main problem with AHP.

In 2007, Karlsson et al. conducted an experiment of comparing tool-supported pair-wise comparison with planning game (PG) [19]. The results show that tool supported pair-wise comparison was less time consuming than PG. While PG seemed to be more difficult to use, its results were found to be slightly more accurate; however, the differences were not statistically significant.

Lethola and Kauppinen conducted an experiment with industry practitioners to evaluate two requirements prioritization methods, pair-wise comparisons and Wiegers' method [21]. The results indicate that prioritization methods may have limited ability to support decision-making in market-driven product development. Moreover, Lethola and Kauppinen identified which prioritization methods are present in the studied companies [21]. The existing methods are: mutual cost-value analysis, modified Kano model, and evaluating aspects affecting to priorities. In addition, product and project level prioritization practices were identified. At project level, grouping requirements, negotiation, and impact validation were used, while at product level, priority lists and open-ended multirelease planning were used.

Herrmann and Paech conducted two quantitative experiments with students to evaluate two requirements prioritization techniques, risk estimation and ranking [13]. The results highlight challenges of risk estimation and what is important during practical requirements prioritization based on risk estimation.

The focus of the above mentioned studies, with the exception of Lethola and Kaupinnen [21], has not been primarily on how QR are prioritized in industry, but focus on the evaluation and comparison of different requirements prioritization techniques. This paper presents a study with the primary focus on how QR are prioritized in industry. Even though Lethola and Kaupinnen [21] identified which requirements prioritization techniques are used in industry, their focus was not on QR.

III. RESEARCH METHODOLOGY

The study was carried out using a qualitative research approach [28]. 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. In addition, qualitative research is directed primarily at collecting and analyzing data with the aim of achieving information depth rather than breadth [3] in an inductive way [7]. The purpose of this study is to gain in-depth understanding of the nature of QR prioritization within

market-driven embedded systems companies. The aim is also to provide a basis for future research. Due to the explorative nature of the study, a qualitative approach has been considered suitable. Furthermore, due to the potential richness and diversity of data that could be collected, semi-structured interviews would best meet the objectives of this study. Semi-structured interviews help to ensure that common information on pre-determined areas is collected, but allow the interviewer to probe deeper where required. In addition, by using semi-structured interviews instead of using a large survey, the interviewer had the chance to validate the questions with the interviewee lessening changes of misunderstandings. The research questions in Table I provided a focus for the empirical investigation.

TABLE I.   RESEARCH QUESTIONS

Research Questions
RQ1: How is QR prioritization performed?
RQ2: What criteria are used when QR are prioritized?
RQ3: How does QR prioritization at product and project levels differ?

A. Research Design and Data Collection

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 analyzed. The investigation can be divided into three phases:

Planning/Selection: The first phase of the study involved brainstorming and planning meetings to design the study and to identify different areas of interests. The sampling strategy used was a combination of maximum variation sampling and convenience sampling within our industrial collaboration network [24]. A "gate-keeper" at each company identified two subjects that he/she thought were the most suitable and representative of the company to participate in this study. Eleven software development companies participated in the study, and from each company, one product manager (PM) and one project leader (PL) from the same project were interviewed, resulting in 22 data points. 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.

All eleven companies develop embedded systems using a market-driven software development approach. The included companies vary in respect to size, type of product, and application domain, a rudimentary characterization (following the guidelines of [14]) can be see in Table II (more details are not revealed for confidentiality reasons).

Data Collection: The study used a semi-structured interview strategy [28]. One interviewee and one interviewer attended all interviews. First, the purpose of the study and a general explanation of QR were presented and then questions about the different areas of interests in relation to QR were discussed in detail. All interviews varied between 40 and 90 minutes.

---

1 http://serg.cs.lth.se/research/experiment_packages/quality_requirements

Analysis: The content analysis [28] involved creating categories where interesting parts from the interviews were marked and discussed. The first two authors examined the categories, first individually and then together in a workshop setting. The category analysis included examination of the content from different perspectives and a search for explicitly stated or concealed pros and cons in relation to prioritization of QR in industry. The results from the analysis are found in Section IV.

B. 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 Wohlin et al. [34].

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 [34] was avoided by collecting data from a wide range of sources on the topic of the study. The potential problem of evaluation apprehension [34] was alleviated by the guarantee of anonymity as to all information divulged during the interviews, and the answers was only to be used by the researcher, i.e. not be showed or used by any other participants, companies, or researcher.

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. In order to obtain highly reliable measures, the interview instrument, including posed questions, two pilot studies were conducted prior to conducting the interviews.

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 potential problem of instrumentation threats was alleviated by developing the research instrument with close reference to literature relating to QR, influenced by previously validated interview instrument [18]. In addition, maturation threats were alleviated by reducing the duration of interview sessions by collecting background information before the interview, and by keeping the interview session to 90 minutes. Selection bias is always present when subjects are not fully randomly sampled. However, interviewees were selected based on their roles by a "gate-keeper" at 11 different companies from different geographical locations, which limited the effect of this threat.

External validity: The external validity is concerned with the ability to generalize the results, i.e. in this case the applicability of the findings beyond the included companies. Qualitative studies rarely attempt to generalize beyond the actual setting since it is more concerned with explaining and understanding the phenomena under study. The nature of qualitative designs also makes it impossible to replicate since identical circumstances cannot be recreated.

	# Employees	Domain	Development process	# of Reqs in a typical project	% of QR in a typical project
A	~100	Control systems	Incremental development	>1000	~10%
B	~3000	Telecom	Plan-driven	~7000	~10%
C	>5000	Telecom	Plan-driven	>20000	Unknown
D	325	Telecom	Agile – Scrum	~100 features	~10%
E	65	Control systems	Waterfall – Iterative	Differs	Differs
F	~700	Surveillance	Iterative	~250	~15%
G	~100	Consumer electronics	Plan-driven	~300	~5%
H	~700	Telecom	Agile – Scrum variant	~200	20%
I	~50	Security	Waterfall and Agile – Scrum	~100	~15%
J	~90	Control systems	Plan-driven	~100	~10%
K	280	Telecom	Waterfall – iterative	~1000	~5%

TABLE II.   COMPANY CHARACTERISTICS

However, understanding the phenomena may help in understanding other cases and situations. The fact that more than one company acknowledges most of the identified challenges increases the possibility to generalize the results beyond this study. The large number of companies and contexts also contributes to generalizability. To avoid the interaction of selection and treatment, interviewees were selected according to their roles within the company by a "gate-keeper", and companies were selected from different geographical locations.

IV.   RESULTS AND ANALYSIS

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

A.   Prioritization of Quality Requirements (RQ1)

In analyzing Research Question 1 (RQ1), this section examines how QR are prioritized in industry. Table III shows what prioritization method was used at the 11 companies (note that Table III displays the answer for both PM and PL, i.e. if only one method is shown for one company, both PM and PL gave the same answer).

TABLE III.   PRIORITIZATION METHODS

Company	Prioritization method(s)
A	Numerical assignment
B	Ad-hoc, Numerical assignment
C	Ad-hoc, Pair-wise comparisons
D	Ad-hoc
E	Ad-hoc
F	Numerical assignment
G	Ad-hoc, Numerical assignment
H	Ad-hoc
I	Ad-hoc
J	Ad-hoc
K	Key performance indicators, Ad-hoc

Looking at Table III, we see that ad-hoc (14 out of 22) is the dominant requirements prioritization method for QR among our case organizations. Ad-hoc is followed by numerical assignment (6 out of 22), pair-wise comparisons (1 out of 22), and key performance indicators (1 out of 22). When we asked the interviewees how QR were prioritized, six answered that ranking (see Section II.A) was used (illustrated as numerical assignment in Table III). However, when the interviewees described how they prioritize QR, the description of the method was similar to numerical assignment (see Section II.A). This interpretation of ranking differs from the one formulated in literature. This leads to a possible mismatch between the established academic interpretation of ranking and the industrial interpretation of it.

The finding that ad-hoc is the most common prioritization "method" of QR is not in line with [4], which stated that numerical assignment is the most common prioritization technique. The unstructured prioritization of QR warrants the question if there is any difference of prioritization of QR between mature as opposed to new products, e.g., unstructured prioritization of QR for new products may be expected as the quality may not be well understood for that product, while mature products may involve a more structured prioritization technique. This is however not the most likely case since the focus of this study is how prioritization of QR is performed. If different techniques and methods had been used for mature and new products, the practitioners would probably have mentioned this. Another question lies in prioritization over time, i.e., prioritization of improved quality over the entire lifecycle for the product as opposed to prioritization in the beginning of the project. There may be such differences, but these were not discovered by this investigation as it only reflects on the method used up-front, not specifically how the QR changes

over time. Life-cycle priority change analysis is an interesting matter for further research.

Interestingly, while [13], [15], [17], [19], [21] have conducted experiments to evaluate which prioritization technique is more promising than others. Out of all the methods tested, only pair-wise comparisons (1 out of 22 in Table III) is used at our studied companies.

One possible explanation for this discrepancy may be explained by the focus, i.e. we focused solely on prioritization of QR, while [13], [15], [17], [19], [21] evaluated prioritization techniques for prioritization of requirements in general. However, as many as 14 out of 22 interviewees indicate that FR are prioritized using the same method as for QR. One explanation may be that time to market is important, and to use a more complex prioritization method takes too long, which was explained by several interviewees. Another possible explanation may be related to customer input. Several interviewees stated that a few important customers are selected to prioritize all requirements for the supplier, despite using a market-driven development approach. No further elaborations were given. Scalability of methods can also be a factor of not using the techniques, which is especially evident with pair-wise comparisons [20].

Ad-hoc prioritization: In total for as many as nine out of eleven case organizations, either both or one of PM and PL indicate that QR are prioritized in an ad-hoc fashion. There are different "approaches" at the studied companies of how QR are prioritized in an ad-hoc fashion: (1) customer input, (2) marketing department, and (3) "gut feeling" (based on the practitioners experience). In three case organizations it is the customer who prioritizes all QR, either direct (provides a list of prioritized requirements to the developing company) or indirect (the PM or PL makes a decision, based on his/her gut feeling/experience, if a particular requirement may be important to their customers). One interviewee explained, *"if a quality requirement is important to the customer, it is a high priority. If it is not important to the customer, it has a low priority, meaning it will not be implemented".* Another interviewee explained, *"In Scrum, we have a direct prioritization by the customers. If quality requirements are not added and prioritized by the customers, we will not improve the quality of the products".*

In some case companies, the marketing department handles the prioritization of all requirements, including QR. However, the software department has the "right" to change the list of priorities based on their experience and believes of what level of quality may be expected by the market and potential customers.

In as many as four companies, the PM or PL's gut feeling is the prioritization method. One interviewee explained, *"some people [PM and PL] think quality requirements are important, while others think release time is more important".* That is, it is up to the PM or PL in charge of a certain product/project to prioritize what is important. Another interviewee further explained, *"some quality requirements (security) are always important for everyone, while others (e.g. usability) are down prioritized by some managers, but some managers consider them important".*

Another factor that affects the decision maker's gut feeling is internal and external stakeholders. The stakeholder who "screams" the loudest is heard, i.e. their preferences are given the highest priority.

Numerical assignment: In literature, numerical assignment is described as putting requirements into different categories based on their importance, e.g. using categories like high, medium and low (see Section II.A). All interviewees, using numerical assignment to prioritize QR, stated that three categories are used when they prioritize QR, which is in line with [32]. Two different sets of categories are used at our studied companies, 1, 2, or 3 (where 1 is the highest priority), or *critical*, *medium*, and *not critical*, which is in line with the results in [15]. However, in three out of four case companies that use numerical assignment, all QR are by default put into the not critical category, i.e. QR always have lower priority than all FR. One interviewee explained, *"once we have implemented all functional requirements, then we look into quality requirements. If, and only if there is time and resources available, then we will prioritize our quality requirements and improve the quality of the product".* In a study by Berntsson Svensson et al, one challenge in managing QR was to get QR into projects when FR are prioritized [5], which is in line with the results in this study.

In the fourth company, QR are by default added to the critical category of requirements. The reason is due to the company's competitors. According to one interviewee, the only way to differentiate from their competitors is to have a product with the highest level of quality, having a competitive advantage.

Prioritization in agile development: Racheva et al. conducted a literature review on agile requirements prioritization methods that can be found in the literature [25]. The result is a list of 15 different requirements prioritization methods; however, none of the found methods in Racheva et al. [25] are used by any of the case organizations (D, H, I) that use Scrum in our study. According to all six interviewees, QR are prioritized in an ad-hoc fashion. One possible explanation to the differences may be explained by the focus, i.e. we focused solely on prioritization of QR. However, the main reason for not using any prioritization method is, according to several interviewees, that *"agile makes projects the focus and not the system, or long term view. This is very detrimental for quality requirements as you do not see the big picture or look beyond your own release".*

General discussion on prioritization challenges: Looking at how QR are prioritized, most of the studied companies do not use a specific method, while four companies use numerical assignment. However, the ones using numerical assignment do not prioritize QR, instead, all QR are either put into the critical or not critical category by default. We asked the interviewees why prioritization of QR is challenging, or why QR are by default put into a preselected category. Surprisingly, not a single challenge or reason was related to the prioritization itself. However, this result is in line with the results in Lethola and Kauppinen [21]. Their results indicate that the challenges in requirements prioritization are other than to order a list with

a set of requirements. In our study, the reasons for difficulties in prioritizing QR are:

- Elicitation of QR
- Lack of well specified QR
- Quantify QR
- What is good enough?
- Knowledge about QR

Elicitation of QR is related to difficulties in identifying the important QR. However, even if the important QR have been elicited, another challenge lies in specifying QR, which affects the prioritization process. One interviewee explained, *"if the quality requirements are well described, they are as easy as functional requirements to prioritize"*. The importance of writing understandable requirements is confirmed in a study by Karlsson et al. [18]. To be able to prioritize QR, quantified (measurable) QR seem to be of major importance. If the managers do not know which level of quality is expected, they cannot prioritize them. Even if QR are quantified, what is good enough asked one interviewee. The interviewee further explained, *"we must be able to understand what the market needs, and to be able to understand the value and the affects level of quality has on the products, but we do not have that experience"*. There seems to be a lack of understanding and knowledge about managing and handling QR, not only in prioritization, but also from elicitation to understanding market needs. All of these challenges affect the prioritization process of QR, which was confirmed by several interviewees.

Several companies have stated that the main challenge with QR is in the elicitation, specification, quantification, and knowledge about QR, and not primarily in the prioritization of them. This is however relevant as the handling of QR is a chain of events, from elicitation down thorough specification and prioritization. This paper focuses on the latter. Even though many companies have inadequately specified (and elicited) QR all companies have QR in one shape or another. Thus, the prioritization of them, in relation to each other and in relation to FR, is important. The fact that companies use ad-hoc prioritization may be explained by several reasons. For example, ad-hoc prioritization may be adequate for the task in most cases in our studied companies. This is however probably not the case as the dismissal rate of QR during project development is 22.5% [5]. Another reason could be that there is inadequate technology and knowledge transfer [9] from research to industry. Also, a likely explanation could be that there is a lack of usable and useful (scalable) prioritization techniques.

General discussion on comparison between functional and quality requirements prioritization: We believe that the main difference in prioritization of QR and FR is that QR have the potential of being measured with a sliding value on a continuous scale rather than being either included or excluded. The quality level is thus typically not viewed as either good or bad, but rather as something with different shades of goodness on a sliding scale, thus adding another dimension to consider in the prioritization process. However, although many prioritization techniques, e.g., the cost-value

approach [16], are mainly used for FR, specific quality targets can of be included as discrete objects of prioritization in these techniques. This may be one reason to why several subjects stated that the same prioritization methods are used for both FR and QR. The prioritization of what level of quality is needed may be viewed by the subjects as specification and quantification, which may explain why these issues were considered the main challenges of prioritization of QR.

B. Used Criteria when QR are prioritized (RQ2)

We asked the interviewees what criteria are taken into account when prioritizing quality requirements. Looking at Table IV, the only criteria mentioned by more than one practitioner were: no criterion (7 out of 22), customer input (6 out of 22), value (4 out of 22), and cost (3 out of 22).

TABLE IV.   CRITERIA USED IN QR PRIORITIZATION

Company	Role	Criteria
A	PM	Cost, Return of investment
	PL	No criterion
B	PM	Customer input
	PL	Value
C	PM	Value
	PL	Value
D	PM	Cost
	PL	Cost
E	PM	No criterion
	PL	No criterion
F	PM	No criterion
	PL	Value, Customer input
G	PM	No criterion
	PL	Importance of quality requirement
H	PM	Customer input
	PL	No criterion
I	PM	Customer input
	PL	Customer input
J	PM	No criterion
	PL	Customer input
K	PM	No criterion
	PL	Cost/benefit

Looking at Figure 1 and 2, almost the same criteria are used when prioritizing QR by numerical assignment or in an ad-hoc fashion. In both prioritization methods, value estimates, cost estimates, and customer input are taken into account. However, when examining how many interviewees stated they used a certain criterion, there was a difference between the two prioritization methods. Only one (the number within parenthesis in Figure 2) interviewee uses customer input when putting QR into different priority groups, while as many as five (see Figure 1) interviewees prioritizing QR in an ad-hoc way took the customer's input into account. One explanation may be, if the customer provides input to, or actually prioritizes the requirements, there is no need for the company to prioritize the QR themselves, which is strongly related to the description of how QR are prioritized in Section III.A.
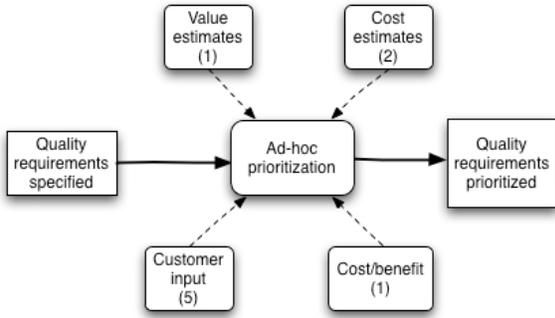
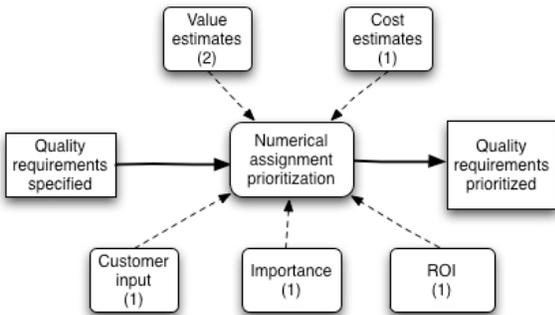Figure 1. Criteria in ad-hoc prioritization

Figure 2. Criteria when prioritizing by numerical assignment

In the study by Berander and Andrews [4], six aspects (criteria) of prioritization are presented;. Importance, penalty, cost, time, risk, and volatility. Our results show that only two (importance and cost) of these six criteria are taken into account when prioritizing QR at our studied companies. According to [4], in practice, it is important to consider multiple criteria, like cost and value, before deciding if a requirement should be implemented directly, later, or not at all. This is not in line with the results in our study, which show only three practitioners use more than one criterion when prioritizing QR.

Although criteria like customer input, cost and value estimations are taken into account when prioritizing QR, in general, the most common criterion is actually having none.

This is illustrated in Table V (note that the *total* column in Table V includes criteria used in other prioritization methods than ad-hoc and numerical assignment).

TABLE V. FREQUENCY OF OCCURRENCE OF CRITERIA

Criteria	Total	Ad-hoc	Numerical Assignment
No criterion	8	5	2
Customer input	6	5	1
Value	4	1	2
Cost	3	2	1
ROI	1	0	1
Cost/benfit	1	1	0
Importance	1	0	1

Instead of calculating different estimates where the input value is based on "gut feeling", most managers at our studied companies prioritize QR based on a "gut feeling" combined with their own experience. In the study by Lethola and Kauppinen, the results indicate difficulties for practitioners to estimate cost and value for requirements [21], which is in line with the findings of our study.

The top two criteria in ad-hoc prioritization are, to have no specific or explicit criterion defined, and customer input, while when numerical assignment is used, the top two are, no specific or explicit criterion defined, and value estimates. Numerical assignment uniquely uses importance and ROI when prioritizing QR. On the other hand, ad-hoc prioritization uniquely takes cost/benefit analysis into account.

Interestingly, only one interviewee of 22 uses cost/benefit criteria when prioritizing QR. The low result for cost/benefit raises the question how useful prioritization techniques for QR like the cost-value approach [16] are in an industrial context. Furthermore, Herrmann and Daneva found 240 papers on requirements prioritization based on cost and benefit estimation [12]. One may ask if researchers are focusing on the correct set of prioritization criteria given the results.

Surprisingly, not a single one (out of six interviewees) of the practitioner from any of the three case organizations in our study estimates value of each requirement and take this into account when prioritizing QR. No further elaboration was given. This result is not in line with [25], [26], who state that the estimation of business value of each requirement is deemed important. The three criteria taken into account in the prioritization process among our case organizations is: (1) customer input (3 out of 6), cost estimation (2 out of 6), and no specific or explicit criterion defined (1 out of 6). In [26], the practitioners agreed that the developers are active in the requirements prioritization process, which is not in line with our findings. Our results show, in two of three companies, the customers prioritize the requirements at product management level, while only one PL indicated that the customers prioritize their requirements on project level.

C. Difference between Product and Project Level (RQ3)

In analyzing Research Question 3 (RQ3), this section examines the difference of prioritization of QR at product and project level respectively, as illustrated in Table VI.

In general, ad-hoc and numerical assignment are the two most common prioritization techniques when prioritizing QR at both product and project level. However, 73% of all PM prioritize in an ad-hoc fashion, while only 55% of the PL. Moreover, 18% of the PM put requirements in to priority groups (numerical assignment), while 36% of the PL used numerical assignment to prioritize QR. This result is partly in line with [21], who found that, at project level prioritization practices, the practitioners put requirement into groups according to their importance. However, in [21], grouping requirements was not present at product management level prioritization practices, which is not in line with our results. The difference between the studies may be explained by the focus, i.e. we focused solely on QR, while in Lethola and Kauppinen the focus was on requirements in general [21]. Moreover, we only focused on companies working with

embedded systems, while Lethola and Kauppinen have a mix of companies working with embedded, interactive, and software systems [21]. In addition, we have 11 case companies compared to seven in Lethola and Kauppinen [21].

Looking at Table VI, the most common criterion for requirements prioritization at product level is no criterion, while at project level, no criterion, customer input, and value are equally common among our case organizations. When comparing product and project level, no criterion is used when prioritizing QR in 45% of the companies at product level, while only 27% at project level. This is a substantial difference and seems to indicate a much larger criterion focus in projects compared to pre-project.

TABLE VI. FREQUENCY OF OCCURRENCE AT PRODUCT AND PROJECT LEVEL

	Product level	Project level
Prioritization Method		
Ad-hoc	8	6
Numerical assignment	2	4
Pair-wise comparisons	0	1
Key performance indicators	1	0
Criteria		
No criterion	5	3
Customer input	3	3
Cost	2	1
Value	1	3
ROI	1	0
Cost/benefit	0	1
Importance	0	1

Product level prioritization: In general, at product management level, the PM prioritizes, mainly based on his/her gut feeling and experience (no criterion in Table VI), QR at a high abstraction level. In general, QR are prioritized early on in the development process where important information, such as used criteria for prioritization, are specified in a document, which in most of the case organizations is an Excel sheet. Interestingly, although the PM is responsible for all projects related to his/her product, the PM do not have an overall picture of how QR affects different projects and other parts of the system. However, QR are prioritized for several projects by the PM.

Project level prioritization: In general, at project level prioritization, a "list from above" (from PM) of prioritized QR is given to the PL. Each area (project) should prioritize QR at a lower level of abstraction (the QR are broken down from high to low abstraction level); however, QR may affect the entire system and most, if not all FR. Despite the knowledge of the affect QR may have, none of the PL had an understanding of the affects. In most of the case organizations, QR were not broken down to a lower abstraction level, hence not prioritized, apart from being assigned to the lowest priority group. The main reason for not breaking down QR is because *"it requires the most skilled managers and developers at project level, which we do not have access to",* according to one interviewee. Lethola and Kauppinen found that negotiation took place in project meetings (at project level), especially when the

project group could not implement all the prioritized requirements [21], which is not in line with our findings. If QR cannot be fulfilled, there is no negotiation process. Instead, the PL use a tool supported system to change the priority of the QR, and add a short explanation, which in most cases is related to time restrictions and available recourses.

V. CONCLUSIONS

This paper presents the result of an empirical study that examines how QR are prioritized in practice at eleven software companies. Data was collected from 11 PMs and 11 PLs, constituting 22 in-depth interviews in total. We do not claim, in any way, that this paper represents an empirical study that is completely representative of the population. There may of course be companies not part of this study that have different ways of working with prioritization of quality requirements. However, it is relevant to observe that among our 11 case companies, many are market leaders in their respective domain.

In relation to RQ1 that asks how QR are prioritized in industry, the findings reveal that: (1) ad-hoc and grouping (numerical assignment) requirements are the dominant methods for prioritizing QR, (2) although numerical assignment is used frequently in QR prioritization at our studied companies, QR are by default often considered to have the lowest priority, and (3) the reasons for not prioritizing QR was not related to the prioritization process itself, but rather how QR were treated in the overall requirements engineering process.

For researchers, a deeper understanding of QR prioritization in industrial practice provides a focus of what to improve/include in supporting techniques for prioritizing QR. Further, a question that is seldom addressed is how sophisticated the methods actually need to be, and what is desirable according to the practitioners? Many techniques and methods that are developed for requirements prioritization are complex, e.g. techniques based on AHP, but as the results show, complex techniques are seldom used in industrial practice. Instead, simple techniques, such as numerical assignment and ad-hoc prioritization, are used in practice. These findings complement the findings of [4], [21].

The findings for RQ2, regarding criteria used for prioritization, we see:

- It is most common to have no specific or explicit criterion defined when prioritizing QR
- Input from customer is more dominant in ad-hoc prioritization than when using numerical assignment
- Only one interviewee mentioned the use of cost/benefit analysis when prioritizing QR

The result that cost and value estimates are not used as input to the prioritization process of QR may be related to their large impact on the entire system, which make the estimates more difficult to take into account. This result is in line with the findings of [21].

For researchers, the understanding of what criteria are used when prioritizing QR enable a focus on what is needed

by practitioners. These findings contradict the findings of [25], [26] with regards to the importance of value estimates when QR are prioritized.

The findings for RQ3, regarding the difference of prioritizing QR at product and project level, reveal that:

- Ad-hoc prioritization is more common at product level compared to project level
- 45% of product managers use no specific criterion when prioritizing QR
- At project level, it is difficult to break down QR to a low enough abstraction level

Difficulties in prioritizing QR at project level may be related to the difficulties of breaking down QR to a lower abstraction level. Therefore, it is important for practitioners to understand, and requires knowledge of how to transform QR to the right abstraction level. The importance of having requirements at the right level of abstraction is in line with [10], [11].

The findings of this paper suggest that there seems to be a lack of knowledge about managing QR. However, the main problem seems to be that QR are by default seen as having a lower priority than FR, and only prioritized if time and resources are available once all FR have been implemented in the coming release. Hence, product management may not be utilizing QR to achieve a competitive advantage, but focusing on FR. Moreover, several of the organizations in this study use the same prioritization method for prioritizing both QR and FR, even though they are very different in nature.

For researchers, the knowledge of the challenges associated with QR prioritization in industry is central. Further, the realization that indirect effects such as difficulties with abstraction level, elicitation of, and specification of, QR can be an issue affecting effective and efficient prioritization, but can also give new avenues of research. Instead of focusing on weather one prioritization technique or method is better than another, the focus could be on improving the practitioners' knowledge and understanding of managing QR as such. It would also be interesting to study how priorities of QR change over the whole product evolution life-cycle.

References

[1] A. Aurum and C. Wohlin, "The fundamental Nature of Requirements Engineering Activities as a Decision-Making Process," Information and Software Technology, vol. 45, Nov. 2003, pp. 945-954, doi:10.1016/S0950-5849(03)00096-X.

[2] A. Aurum and C. Wohlin, Engineering and Managing Software Requirements (eds.). New Yrok: Springer, 2005.

[3] L. Blaxter, C. Hughes, and M. Tight, How to Research. Open University Press, 2001.

[4] P. Berander and A. Andrews, "Requirements Prioritization," in Engineering and Managing Software Requirements. A. Aurum and C. Wohlin, Eds. Berlin, Springer Verlag, pp. 69-94, 2005.

[5] R. Berntsson Svensson, T. Gorschek, and B. Regnell, "Quality Requirements in Practice: An Interview Study in Requirements Engineering for Embedded Systems," Proc. 15th International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ 09), Springer Verlag, Jun. 2009, pp. 218-232, doi:10.1007/978-3-642-020506-_19.

[6] R. Berntsson Svensson, M. Höst, and B. Regnell, "Managing Quality Requirements: A Systematic Review," Proc. 36th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA 10), IEEE Comput. Soc. Press, Sep. 2010, pp. 261-268, doi:10.1109/SEAA.2010.55.

[7] B. Fitzgerald, "An empirical investigation into the adoption of systems development methodologies," Information and Management, vol. 34, Dec. 1998, pp. 317-328, doi:10.1016/S0378-7206(98)00072-X.

[8] T. Gorschek and A. Davis, "Requirements Engineering: In Search of the Dependent Variables," Information and Software Technology, vol. 50, Jan. 2008, pp. 67-75, doi:10.1016/infsof.2007.10.003.

[9] T. Gorschek, P. Garre, S. Larsson and C. Wohlin, "A Model for Technology Transfer in Practice", IEEE Software, vol. 23, Nov.-Dec. 2006, pp. 88-95, doi:10.1109/MS.2006.147.

[10] T. Gorschek, P. Garre, S. Larsson, and C. Wohlin, "Industry Evaluation of the Requirements Abstraction Model," Requirements Engineering Jurnal, vol. 12, Jul. 2007, pp. 163-190, doi:10.1007/s00766-007-0047-z.

[11] T. Gorschek and C. Wohlin, "Requirements Abstraction Model," Requirements Engineering Journal, vol. 11, 2006, pp. 79-101, doi:10.1007/s00766-005-0020-7.

[12] A. Herrmann and M. Daneva, "Requirements Prioritization Based on Benefit and Cost Prediction: An Agenda for Future Research," Proc. 16th IEEE International Requirements Engineering Conference (RE 08), IEEE, Sep. 2008, pp. 125-134, doi:10.1109/RE.2008.48.

[13] A. Herrmann and B. Paech, "Practical challenges of requirements prioritization based on risk estimation," Empirical Software Engineering, vol. 14, 2009, pp. 644-684, doi:10.1007/s10664-009-9105-0.

[14] M. Ivarsson and T. Gorschek, "A Method for Evaluating Rigor and Industrial Relevance of Technology Evaluations," Empirical Software Engineering, in print, doi:10.1007/s10664-010-9146-4.

[15] J. Karlsson, "Software requirements prioritizing," Proc. 2nd IEEE International Conference on Requirements Engineering (RE 96), IEEE Comput. Soc. Press, Apr. 1996, pp. 110-116, doi:10.1109/ICRE1996.491435.

[16] J. Karlsson and K. Ryan, "A cost-value approach for prioritizing requirements," IEEE Software, vol. 14, Sep-Oct. 1997, pp. 67-74, doi:10.1109/52.605933.

[17] J. Karlsson, C. Wohlin, and B. Regnell, "An evaluation of methods for prioritizing software requirements," Information and Software Technology, vol. 39, Feb. 1998, pp. 939-947, doi:10.1016/S0950-5849(97)00053-0.

[18] L. Karlsson, Å.G. Dahlstedt, B. Regnell, J. Natt och dag, and A. Persson, "Requirements engineering challenges in market-driven software development – An interview study with practitioners," Information and Software Technology, vol. 49, Feb. 2007, pp. 588-604, doi:10.1016/j.infsof.2007.02.008.

[19] L. Karlsson, T. Thelin, B. Regnell, and P. Berander, "Pair-wise comparisons versus planning game partitioning – experiments on requirements prioritization techniques," Empirical Software Engineering, vol. 12, Feb. 2007, pp. 3-33, doi:10.1007/s10664-006-7240-4.

[20] L. Lehtola and M. Kauppinen, "Empirical evaluation of two requirements prioritization methods in product development projects," Proc. European Software Process Improvement Conference (EuroSPI 04), Springer-Verlag, Nov. 2004, pp. 161-170.

[21] L. Lehtola and M. Kauppinen, "Suitability of Requirements Prioritization Methods for Market-driven Software Product Development," Software Process Improvement and Practice, vol. 11, Jan-Feb. 2006, pp. 7-19, doi: 10.1002/spip.249.

[22] A. Ngo-The and G. Ruhe, "Decision support in requirements engineering," in Engineering and Managing Software Requirements. A. Aurum and C. Wohlin, Eds. Berlin, Springer Verlag, pp. 267-286, 2005.

[23] T. Olsson, R. Berntsson Svensson, and B. Regnell, "Non-functional requirements metrics in practice– an empirical document analysis," Workshop on Measuring Requirements for Project and Product Success, 2007.

[24] M.Q. Patton, Qualitative Research and Evaluation Methods. USA: Sage Publications, 2002.

[25] Z. Racheva, M. Daneva, and L. Buglione, "Supporting the Dynamic Reprioritization of Requirements in Agile Development of Software Products," Proc. 2nd International Workshop on Software Product Management (IWSPM 08), IEEE, Sep. 2008, doi:10.1109/IWSPM.2008.7.

[26] Z. Racheva, M. Daneva, K. Sikkel, R. Wieringa, and A. Herrmann, "Do we Know Enough about Requirements Prioritization in Agile Projects: Insight from a Case Study," Proc. 18th IEEE International Requirements Engineering Conference (RE 10), IEEE, Sep. 2010, pp. 147-156, doi:10.1109/RE.2010.27.

[27] B. Regnell, M. Höst, J. Natt och Dag, P. Beremark, and T. Hjelm, "An industrial case study on distributed prioritization in market-driven requirements engineering for packaged software requirements," Requirements Engineering, vol. 6, 2001, pp. 51-62, doi:10.1007/s007660170015.

[28] C. Robson, Real World Research. Oxford: Blackwell, 2002.

[29] G. Ruhe, "Software engineering decision support – a new paradigm for learning software organizations," Proc. 4th International Workshop on Advances in Learning Software Organization, Springer-Verlag, Aug. 2002, pp. 104-113.

[30] G. Ruhe, A. Eberlein, and D. Pfahl, "Quantitative WinWin — A new method for decision support in requirements negotiation," Proc. 14th International Conference on Software Engineering and Knowledge Engineering (SEKE 02), ACM, Jul. 2002, pp. 159-166, doi:10.1145/568760.568789.

[31] T.L. Saaty, The Analytic Hierarchy Process. New York: McGraw-Hill, 1980.

[32] I. Sommerville and P. Sawyer, Requirements engineering – A good practice guide. Chichester: John Wiley and Sons, 1997.

[33] R. Wiegers, Software Requirements. Redmond: Microsoft Press, 1999.

[34] C. Wohlin, P. Runeson, M. Höst, C. Ohlson, B. Regnell, and A. Wesslén, Experimentation in Software Engineering: An Introduction. Boston: Kluwer Academic, 2000.