NTNU
Norwegian University of
Science and Technology

Decision support material to incorporate quality requirements technologies: a systematic literature review and industrial interviews

Phani Saripalli

Master in Information Systems
Submission date: October 2010
Supervisor: Tor Stålhane, IDI

Norwegian University of Science and Technology
Department of Computer and Information Science

Problem Description

Requirements engineering is an important activity in software development. Quality requirements, also known as non-functional requirements are often overlooked and poorly handled. Research also shows a lack of established methods and challenges to handle quality requirements.
The purpose of the thesis is to review the state of the art on quality requirements, grade the papers reviewed to assess research quality and provide decision support material to practitioners to handle quality requirements in software projects. Also, a short survey will be conducted through interviews to identify industrial practices in handling quality requirements and provide a discussion on the merits of research and impact on industrial practices.

Assignment given: 02. April 2010
Supervisor: Tor Stålhane, IDI

NTNU
Norwegian University of
Science and Technology

Decision support material to incorporate quality requirements technologies: a systematic literature review and industrial interviews

Saripalli Phani Shashank

Submitted to**:** Department of Computer and Information Science, IDI

Submission Date: August 2010

University supervisor: Dr. Tor Stålhane

Problem description

Requirements engineering is an important activity in software development. Quality requirements, also known as non-functional requirements are often overlooked and poorly handled. Research also shows a lack of established methods and challenges to handle quality requirements.

The purpose of the thesis is to review the state of the art on quality requirements, grade the papers reviewed to assess research quality and provide decision support material to practitioners to handle quality requirements in software projects. Also, a short survey will be conducted through interviews to identify industrial practices in handling quality requirements and provide a discussion on the merits of research and impact on industrial practices.

Assignment given: 2, April 2010

Supervisor: Tor Stålhane, IDI

Abstract

Background: In an applied engineering field like requirements engineering, the final goal of the research is its adoption by industry. For technology transfer to be possible practitioners need strong evidence for a technology's applicability and possible benefits. Therefore research on a technology should be done in a realistic setting so that practitioners can use the results to build a strong case for its adoption.

Quality requirements are constraints placed on the software. Empirical results show challenges in handling quality requirements, e.g. include late discovery of quality requirements. In addition, we lack technologies that can be used to incorporate them into the final software. Functional and quality requirements differ in nature; treatment given to functional requirements will not always be applicable for quality requirements. Therefore practitioners need decision support material based on empirical evidence to incorporate the suggested technologies.

Objective: The purpose of the thesis is to identify technologies for quality requirements that have been empirically evaluated. Methods in relation to elicitation, specification, metrics (or measurement) and testing will be identified with the aim of providing decision support material to practitioners for incorporating quality requirements in the software. To find possible future directions of requirements engineering research, the current state of technology adoption for quality requirements will be identified.

Method: The presented research is explorative and investigative in nature. A systematic literature review method was employed to identify potential technologies for adoption. An empirical study was conducted with three participants from three companies to get an insight into the state of technology adoption.

Results: The systematic literature review includes 46 papers published between 2000 and 2010. All in all only four of the 46 papers offer high realism and support for technology adoption, i.e. presenting evaluations in a realistic setting, with practitioners using real world industrial applications. Another three papers were found to have potential in terms of furthering technology transfer. A general finding common for many of the papers reviewed is a lack of scientific rigour which affects the credibility of the results. Among the participants in the three companies interviewed, none of the methods presented were used.

Conclusions: Technology transfer support for quality requirements technologies is challenged by low strength of evidence. Evaluations of technologies lack descriptions of evaluation design - description of how evaluations have been performed - and validity. There is a need for more and better empirical evaluations of technologies to handle quality requirements. The results of industrial interviews shows the need for more empirical investigations, for example surveys, to identify current industrial practices and technologies able to handle or incorporate quality requirements in projects successfully.

Keywords Systematic literature review, Industrial interviews, Technology transfer, Quality requirements

Acknowledgments

This thesis has been a result of one semester work, from April to September, 2010 at Norwegian University of Science and Technology (NTNU), Norway.

First and foremost my sincere thanks to my supervisor Prof. Tor Stålhane for accepting me as his master student. His guidance and support in thesis have been invaluable. I thank him for providing industry contacts needed for the interviews. Most of all, I appreciate his patience in going through my work, finding scope for further improvement and assuring me of my work and inspiring me. Thesis meetings have been a big source of confidence. I think I learnt a lot from Prof. Tor, thank you.

I would to thank the interview participants, Miss Ellie Tzatzani from Greece, Mr Trond Johansen from Norway and the anonymous participant from Norway. It has been a pleasure working with you for my thesis.

Credit must be given to my parents for supporting me, to my friends in India and Sweden for encouraging me. Special thanks to Mohanarajesh Kommineni for his effort in indexing and taking care of my indentations and making the work a neat presentation.

Finally I'd extend my sincere gratitude to Mr Gurudutt Velpula, my International coordinator at BTH, Sweden for arranging the exchange studies and encouraging me to proceed with my Master thesis at NTNU.

Table of contents

CHAPTER 1
Introduction

Introduction

Software is an important part of our daily life and an increasingly important factor for all industries [94]. Software today is inevitable in goods like cars, washing machines, video games etc. Software development is a human centred and knowledge intensive activity [48]. Over time, there has been an increase in the attention on software development by researchers and practitioners [95, 96], which can be attributed to the increased use of software. To improve competitive advantage and also to sustain it software organizations should continuously improve their processes and practices [49]. A good Requirements Engineering (RE) process is a success factor for all software products. RE is concerned with identifying the goals for a proposed system and how to convert them into services and constraints [50]. Good RE is critical when designing software and is a major determinant of software quality [50]. RE will help us selecting a requirements subset from an identified superset of candidate requirements so that the stakeholders' system constraints are fulfilled and thus maximizes business value [111]. Errors caused due to requirements are time consuming and expensive to correct [50]. RE is a well established research area [56] that focuses on identifying attributes or features needed for a software product to meet the customers' expectations and needs [51, 57] and often spans the entire software development life-cycle [50]. Improving RE practices is an important step in the overall success of software products [50, 57, 110].

Software projects use RE to elicit, document and manage requirements throughout the project [50]. Requirements are descriptions of how a software system should behave [50] and are classified into two types: functional requirements and non-functional requirements [50]. Functional requirements describe the functionalities of the system - "what the system will do" [50]. Non-functional requirements (NFR) are the constraints put on the types of solutions that will meet the functional requirements [50]. These are also referred to as attributes of the system [56]. Some examples of NFRs, also known as Quality Requirement(s) (QR) [58] are performance, security and maintainability.

Inadequate requirements and poor RE often result in deficiencies in software development [51] such as cost overruns, and improper selection of system architecture [50]. Selection of system architecture is tightly dependent on the selected requirements set [102, 112]. As it is a common practice to estimate software size, schedules and budget in software development from the requirements document [63], a strong focus on RE is inevitable. In addition, it has been observed that software quality strongly depends on RE [50, 52, 53, 54]. Software quality can be defined as a desired combination of attributes in the product [55]. NFRs focus on "how good" software does a task as opposed to functional requirements which focus on "what" software does [57]. The focus in RE usually is on functional requirements [58]. However, the complexity of a system depends both on the functional requirements and NFR [62]. Research has shown that QR is difficult to understand [58] and is rarely distinguished from functional requirements [58, 59, 60]. The importance of understanding QR in industrial context and the need for industrial practices to handle QR in software projects was highlighted by Landes [61]. Not dealing with QR efficiently or not giving enough attention to the importance to QR may result in increased cost of development and time-to-market [9]. Svensson et al. [41] conducted a study to identify the challenges concerning selection and management of QR. A major challenge identified in this study is how to achieve testable QR, i.e. to make well specified and quantified QR. The finding of [41], i.e. challenge of achieving testable QR is in line with observations in [64, 65]. A large amount of effort is invested in defining and implementing functional requirements. The limited focus on QR is a problem area in RE. The importance of aligning RE and testing is being researched as part of EASE project by researchers at Lund University and Blekinge Institute of Technology [66]. The advantage of such an alignment is that in this way testable requirements become an integrated part of the project and will thus be implemented. This alignment is further stressed by Uusitalo et al. [67] and Post et al. [68] who claim that linking requirements and software verification is a key activity in software development.

When searching for new technologies, practitioners need sound decision support [49]. Technologies can be techniques, methods, models, processes and tools [49]. Practitioners need to find strong evidence of the use of technology, it's possible benefits and limitations. In an engineering field like RE [49], adoption of technologies in industry depends heavily on research and research results [69]. However, research in RE is criticized for having little impact on RE practices adoption [70, 71] and not providing technologies that are useful in real environments [49, 72]. This is attributed to the lack of proper evaluation [73]. These claims are not one-sided and research results of RE were also found to be beneficial to practitioners [74] - e.g. scenario-based requirements engineering, agent-oriented requirements and goal modeling.

Evidence-based Software Engineering (EBSE) methodology strives to support and improve technology adoption decisions based on best evidence from research and practical experience. Technology creation stems from a business need or a technical problem [75] and the ultimate aim of RE research is to transfer research results to industry. To facilitate technology transfer research results need to be convincing to the practitioners. Gorschek et al. observed [78] that for technology transfers to be possible management should identify the benefits of research and technology transfer. Lack of management support in technology transfer has been observed by several researchers [113, 114]. While both practitioners and project leaders are concerned with the research results, managements' focus is on the effect of software process improvement on cost, effort, time-to-market and return-on-investment. Lack of evidence of in this area will be a negative factor for technology transfer and adoption. Therefore for technology transfer to be feasible, practitioners need decision support in terms of strong evidence from research results in addition to software process improvement initiative aspects that are considered by the management when making decisions. However, the management aspect of decision support is out of the scope of this thesis and is not considered further. Instead, we focus on providing decision support to practitioners based on best evidence from research and practical experience according to the EBSE paradigm.

1.1 Purpose Statement

The purpose of the thesis is to identify methods for quality requirements that have been empirically evaluated. Methods in relation to elicitation, specification, or measurement and testing will be identified with the aim of providing decision support material for practitioners when deciding whether to incorporate methods for handling quality requirements in the software. The possibilities of technology transfer of the evaluated technologies will be assessed by investigating studies that offer a high degree of realism. To identify the usefulness of adoption, strength of evidence of the reported studies will be evaluated. Finally we will assess the state of technology adoption of quality requirements research in the industry.

1.2 Research outcomes

The contribution of this research is a report documenting the following:

- A list of quality requirements-related technologies evaluated in industry and academia for elicitation, prioritization, specification, measurement and testing.
- Priorities/views/attitudes of practitioners on incorporating quality requirements as found in research (academia).
- The degree of realism offered by the identified empirical evaluations.
- A list of technologies that have the possibility of adoption for practitioners.
- A list of quality requirements that have been implemented in real-world industry practice.
- A list of technologies adopted in practice.

CHAPTER 2

Background and Related Work

This chapter gives an introduction to Quality Requirements (QR), technology transfer and decision support based on evidence plus related work in the area of QR.

2.1 Background

Requirements can be classified as functional requirements or non-functional requirements [50, 51]. Non-functional requirements are also known as Quality Requirements (QR) [97, 98]. Functional requirements can be defined as *"Requirements that specify the functions of the system, how it records, compute, transforms, and transmits data"* [99]. Functional requirements specify "what the system will do" [50] while QR is defined as the constraints put on the types of solutions that will meet the functional requirements [50]. QR describes how software does a particular task [99, 101] and is relevant to system properties such as reliability and performance [102]. These are also known as soft goals [62].QR put restrictions on the system and limit the choice for constructing solutions to the problem [102]. Not dealing with QR effectively or not addressing QR in the design of solutions leads to errors that are difficult to correct at a later stage of software development [103, 104. 106]. The importance of QR and addressing it early in the projects is addressed in [105]. The *i** framework used agent and goal-oriented approaches to address RE issues modeling and analyzing stakeholders' interests and organizational characteristics was proposed and validated in a real setting by Yu [105]. The results of the empirical validation were positive.

There are several standards that classify QR – e.g. ISO/IEC 9126 and IEEE 830-1998 [99]. Quality models like Boehm's Quality model is also used for classification of QR. Quality models have quality characteristics which are further divided into sub-characteristics. For example, ISO/IEC 9126 has six quality characteristics, namely functionality, reliability, usability, efficiency, performance and portability. These six characteristics are further classified into 27 sub-characteristics. Chung et al. [108] claim that there is no fixed set of quality characteristics and the terminology of QR varies a lot among practitioners, researchers and the general public. This makes identifying and implementing quality characteristics for software products difficult.

2.2 Related work

In a recent study based on a survey [8] it was observed that development tools used by practitioners are not well suited for NFR. Also, NFRs are not well utilized and seldom used for making architectural and technical decisions. This is in line with the studies [58, 60] where implementation and utilization of QR was found to be difficult. There has been QR-research but only piecemeal. Lubars et al. [64] conducted a survey on requirements modelling. The key observation from this study is that requirements mostly are vaguely specified, which causes difficulty in understanding and implementing them. This was found to be a major challenge for specifying performance requirements also [64]. It has also been observed that requirements are difficult to test as they are not clearly stated [65].

To discover QR late in the product development cycle causes many of the software development problems [103, 104, 108, 109]. A need for methods to assess the quality characteristics in software was identified by Chung et al. [108]. Borg et al. [109], in their empirical study involving two software developing organizations in Sweden found some of the problems associated with QR are that QRs are stated in non-measurable terms and many QR remain undiscovered. These problems were associated with a lack of methods for handling QR thus identifying a need for methods and tools that support QR throughout the entire development process.

Cysneiros et al. [104] investigated a strategy that allows integration of QR into data models using entity relationship diagrams thus providing a means to specify and analyze QR. Kamsties et al. [65] also observed challenges related to specifying QR and more specifically usability requirements. This is in line with the result of the survey [41], where the major challenge was found to be how to specify testable QR. The difficulty in specifying QR stems from the fact that most functional requirements and QR are intertwined [41]. Therefore in most cases QR are neglected or passed over too easy due to the lack of methods. To address this issue methods like NFR Framework and i* family were [62] proposed.

There have been several reviews of RE - e.g. [49, 77, 79, 115] and also on software engineering topics like software cost estimation [116] and agile methods [87]. Reviews in software engineering so far have been summarized by Kitchenham et al. [80]. Some of the software engineering topics that were reviewed include cost estimation, testing techniques, COTS development methods, experiments in Software Engineering and Software architecture evaluations methods. Reviews specific to QR has not yet been done. Earlier reviews in RE – summarized below - did not cover the perspective of QR. Davis et al. [77] conducted a systematic review of the effectiveness of requirements elicitation techniques. The following summarize the results of this study:

- Structured interviews appear to be one of the most effective elicitation techniques and are suitable in a variety of domains.
- Techniques like card sorting, prototypes and ranking tend to be less effective than interviews.
- Analyst experience or expert judgement does not appear to be relevant factors during information gathering using interviews. Experts and novice analysts were found to be equally good in information acquisition.

Although interviews are people-oriented rather than scientific in nature, it is a widely used for elicitation.

Concepts and techniques in RE were reviewed by Lamsweerde [78] who observed the importance and prominence of requirements modelling using goal-based approaches rather than object-oriented methods. Goals provide a basis for the analysis of conflicting requirements. This later led to the development of goal-based reasoning as a framework to model and analyze NFR [62].

Ivarsson [49] reviewed technology transfer decision support in requirements engineering research based on articles published in Requirements Engineering journal (REj). Key findings from this study are that few studies presented in the review provide strong support for methods' applicability and many of the evaluations presented were not done in realistic settings. The studies were either done using toy examples invented for the purpose of evaluations or used researchers or students as subjects for evaluations. The evaluations suffered from lack of realism and thus provided limiting information that could be used to decide whether to use the techniques or not. Some of the evaluations that provide support for applicability of methods include Requirements Abstraction Model (RAM) and Knowledge-based Approach for the Selection of Requirements Engineering Techniques (KASRET).

Parviainen [79] presented an inventory of RE technologies, mainly from the perspective of embedded systems. Key findings of this study are that new technologies are often not mature enough to be applied in real-world applications and majority of technologies are not known to the practitioners. It was found that companies need guidance on how to find information on technologies and their possible benefits, limitations and suitability to their needs. Although an inventory of technologies was presented, the study does not clearly describe the literature survey method. Nevertheless, this study showed a clear need for support for practitioners in identifying technologies that will suit industrial needs.

The gap between research in RE and industrial practices has been observed in [70, 71]. Technology transfer, the process of moving new technologies from academia or research centres to industry [49] needs management support [74, 78] and strong evidence [76]. Several models for technology transfer have been proposed - e.g. [72, 77, 78]. A general model for technology transfer was developed by Gorschek et al. [78].

This thesis reviews research related to QR from a technology transfer perspective by adopting Kitchenham's SLR [81]. The motivation for the thesis is lack of reviews on QR and decision support material for technology adoption. The motivation of adopting SLR is that it is systematic and allows careful synthesis of research available. The core idea of technology transfer and decision support is heavily inspired by [49, 79].

CHAPTER 3
Research Agenda

3.1 Research Focus

The focus of this research is on reviewing articles that present evaluations of technologies for handling Quality Requirements (QR) in software projects. Technologies addressing the following five process areas will be reviewed: elicitation, prioritization, measurement (metrics), specification and testing. The review process aims at identifying technologies, classifying the problems process areas they address and provide decision support material to practitioners for adoption of technologies in industry practice. An additional focus of the research is to identify technologies used in the current industry practices.

3.2 Objectives

The following objectives are central to this thesis:

- To identify the priorities in selecting/opting quality requirements.
- To identify methods for elicitation and prioritization of quality requirements.
- To identify technologies for quality requirements in relation to specification, measurement (metrics, quantification) and testing.
- To evaluate the rigidity of the research results so as to assess the strength of the presented evidence.
- To identify decision support, based on strength of evidence.
- To conduct a brief survey so as to identify real life industry practices in relation to the above mentioned objectives.

3.3 Research Questions

Research questions (RQ) are formulated based on the objectives and are shown in Table 1.

Table 1 Research questions for the thesis

Research questions
RQ1: What RE technologies exist for QR?
RQ1.1: What are the most selected QRs?

RQ 1.2: What themes are addressed by studies on QR elicitation, prioritization, specification, metrics and testing?

RQ2: What is the current state of evaluation of RE for QR?

RQ2.1: What is the realism found in the studies reported?

RQ3: What can be inferred from the studies for the benefit of implementation of QR?

RQ3.1: What is the strength of evidence available in the reported studies?

3.4 Research method(s)

Research methods provide procedures and guidelines for investigations [82]. The utilization of research methods is important in all research. Research methods guide researchers in identifying the process of investigation needed to answer the research questions. Some of the commonly used research methods are systematic literature reviews (SLR) or systematic reviews [81], surveys [84], case studies [85] and experiments [86]. In this thesis Qualitative research methodology [82, 83] is used together with SLR and industrial interviews.

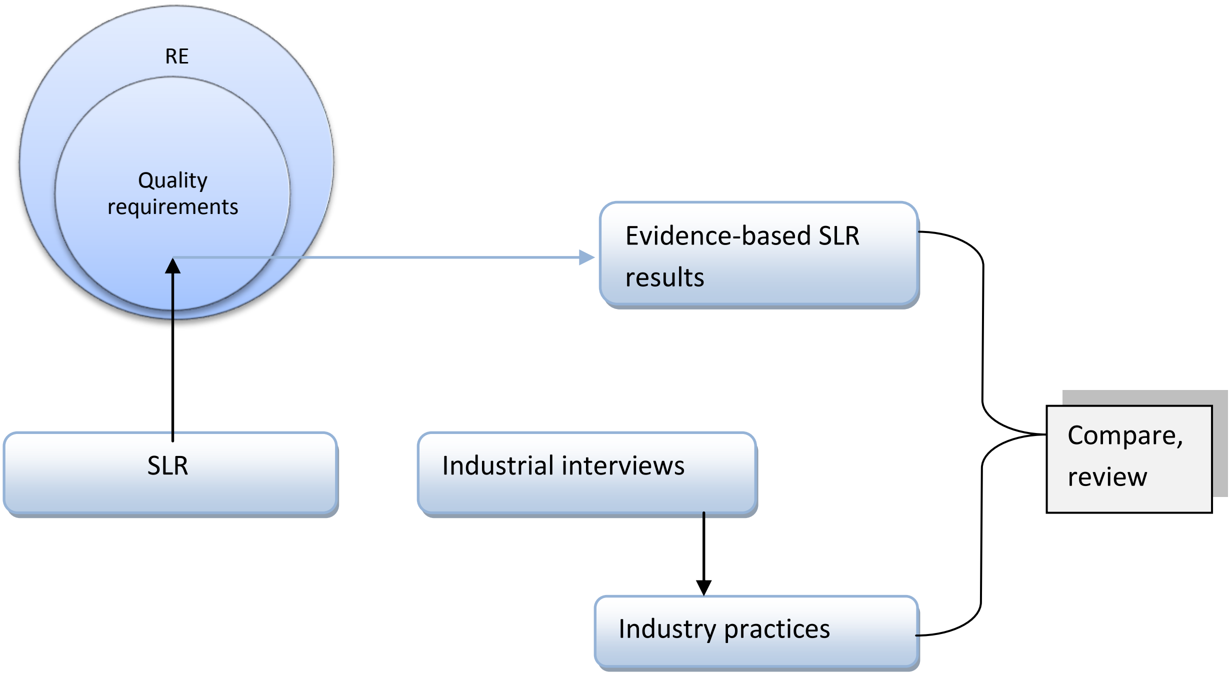
The thesis is explorative and investigative in nature. As discussed earlier, the thesis aims at investigating and identifying decision support available for quality requirements technologies. This is done at two levels. At the first level the available peer-reviewed articles were reviewed to identify technologies that have been evaluated. At the second level industrial interviews were conducted so as to identify technologies that are adopted and used in practice.

As mentioned earlier, this thesis attempts to explore and identify technologies so as to offer decision support to practitioners if they are looking for a QR method. The thesis also intends to identify research gaps and areas of further investigation for research and sum up the available knowledge in a systematic manner. To this end SLR is used. Systematic reviews have received increasing attention in software engineering [49]. SLR is a systematic approach to identify, evaluate and interpret research available about a particular area of interest [81]. Systematic reviews evaluate and interpret available research related to a research question or subject of interest [81] in evidence-based software engineering [80]. It is a structured and repeatable process with a defined search strategy to collect studies relevant to the goals of the review.

Use of a documented search strategy allows researchers to identify of primary studies and to perform replication of the review [37]. SLR allows an unbiased approach in defining a review protocol that can be used to identify peer-reviewed literature published across several search venues, i.e. publication journals and conference proceedings. This makes the SLR systematic as opposed to an ad-hoc process. Study selection criteria are defined and used to identify studies relevant to the purpose of the review. An example of study selection criterion is found in the review on Agile methods [87]. As the thesis primarily aims at exploring technologies related to QR instead of understanding a particular case or comparison of technologies, other methods such as case studies or experiments are not appropriate and thus not used.

In software engineering research there is trend towards using empirical research methods like case studies, surveys, experiments etc. depending upon the objectives of the study [93]. Empirical methods are important and should be used to bridge the gap between academia and industry [76]. This is because empirical methods can be used to explore and understand real-world industry problems by cooperating with practitioners and thereby providing researchers in academia to develop technologies addressing practitioners' concerns. For example, a technology invented in academia can be tested in a pilot project in industry, showing the possible benefits of the technology. Also, possible limitations of the technology found in the pilot project can later be rectified. Software engineering – as all other engineering fields – is a combination of technical aspects and social factors [86], the use of empirical methods is prominent.

Figure 1 Research methodology for the thesis



As part of the investigation, a survey was used as the second research method. A survey studies a phenomenon in a population by studying, or surveying a sample that is assumed to be representative of that population. This can be done by collecting qualitative data through questionnaires and interviews. In this thesis, we have done a short survey by interviewing people from the software industry. The aim of the industrial interviews is to explore and understand real-life industrial practices, in relation to the aim of the thesis. Figure 1 outlines the research methodology used in the thesis.

CHAPTER 4

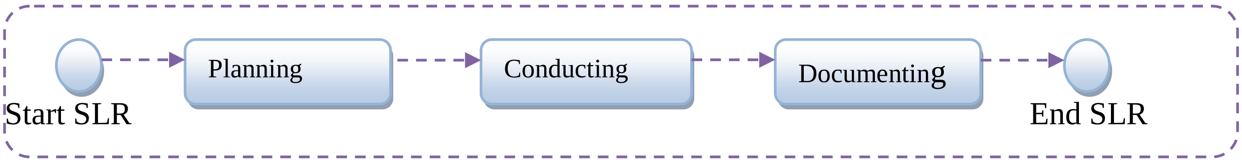
Systematic literature review (SLR)

A systematic literature review (SLR) is a way of identifying, evaluating and interpreting all available research relevant to a particular research question, topic, or phenomenon of interest [81]. Some of the factors that make SLR different from a normal literature review are outlined by Kitchenham [81]:

- SLR defines a review protocol specifying research questions and search venues
- SLR involves explicitly documenting the search strategy and review procedure so that reader can assess its rigor and completeness and a replication of the review is possible in the future
- SLR involves having inclusion/exclusion criteria that aid in identifying studies relevant to the purpose of the review
- SLR involves evaluating the quality of the reported studies

SLR consists of three main phases: planning the review, conducting the review and documenting the review. The review process is outlined in Figure 2. In the planning phase, the need for a SLR is identified and a review protocol or design is developed. Review design includes identifying objectives of the review, defining research questions, selecting a search strategy for identifying primary studies relevant to research questions, inclusions and exclusion criteria, quality assessment of the studies reported and a strategy for extracting data from the reported studies [81]. The aim of selecting a search strategy is to reduce the potential bias of the researcher so that future replication of the review is possible [81].

Figure 2 SLR phases (adapted from [81])



The second phase is to conduct the review, i.e. identifying primary studies in the search venues and data extraction from these primary studies. For the purpose of data extraction, data extraction forms are used. In the last phase, extracted data are analyzed and documented.

The subsequent sections describe the review process developed and used in this thesis. The process is based on the guidelines developed by Kitchenham [80] with the difference that in this thesis study quality assessment is combined with inclusion/exclusion criteria. In this way we can assess the quality of a study based upon the empirical value it gives to the research community in Software Engineering. As the purpose in this thesis is to evaluate and appraise empirical studies and studies with empirical evaluations, we do not include pure research papers. Studies with exclusive focus on QR, in our case, have better value rather than studies which are focused on functional requirements. To cover these two aspects, quality assessment is included in inclusion/exclusion of the studies. Although quality assessment begins in parallel with the selection of studies, studies are evaluated based scientific rigour. The process of evaluating studies is explained in Chapter 6.

To this end the process of identification of studies and data extraction are adopted from [49]. The author draws inspiration for combining inclusion, exclusion criteria and study quality assessment from [49], as this provides a comprehensive way of answering RQ3.

4.1 Identification of the need for review

Kitchenham et al. [80] have reported several systematic reviews in software engineering but a review relevant to QR is missing in the reviews reported to date. Moreover, to identify similar work, the author performed a search in Inspec/Compendex with the search string:

("quality requirements" OR "non-functional requirements" OR "non functional requirements") AND ("systematic literature review" OR "systematic review")

Neither the studies reported through the above search string nor the studies reported in [80] addressed the research questions of the thesis.

4.2 Research questions

The research questions are formulated in Chapter 3. Table 2 shows the research questions together with a short description.

Table 2 Research questions (RQ) and description

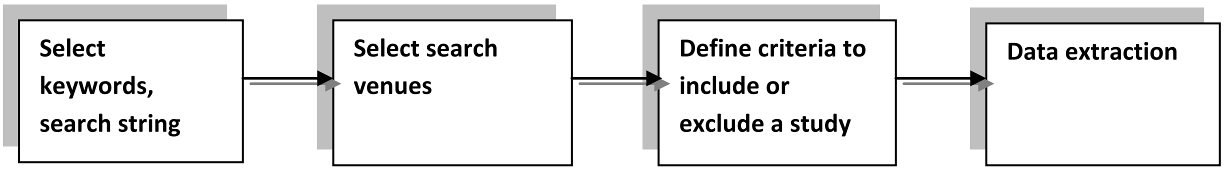
Research questions (Quality requirements = QR)	Description
RQ1: What RE technologies exist for QR?	To identify the views and attitudes of practitioners in selecting quality requirements for software. An inventory of technologies in relation to QR elicitation, prioritization, specification, metrics and testing will be identified.
RQ1.1: What are the most selected/opted QRs?	
RQ 1.2: What themes are addressed by studies on QR elicitation, prioritization, specification, metrics and testing?	
RQ2: What is the current state of evaluation of RE for QR?	The delimitation factor in the review is "empirical studies" reported in relation to QR. The realism found in the evaluations performed within QR will be identified.

	This will be done by observing research methods utilized, research context, subjects involved in the study and the scale of the application used in the reported studies.
RQ2.1: What is the realism found in the studies reported?	
RQ3: What can be inferred from the studies for the benefit of implementation of QR?	The strength of evidence available in relation to QR, can be found in the reported studies. To identify the evidence, the presentation of the study is important. Thus, the context of the study should be described thoroughly so that the results can be transferred to another environment. The validity of study is important and should be described.
RQ3.1: What is the strength of evidence available in the reported studies?	

4.3 Search strategy

The purpose of a search strategy is to have a systematic process for searching studies relevant to the research questions in the defined search venues. Identification of primary studies should be unbiased [49]. Figure 3 outlines the search strategy used in this thesis.

Figure 3 Search strategy


Select keywords, search string
Select search venues
Define criteria to include or exclude a study
Data extraction

4.4 Search string

Several keywords were identified and elaborated based on trial searches. The motivation for conducting trial searches is the large amount of keywords and terminologies used in software engineering. It is therefore important to identify the keywords that are relevant for the current research when formulating the search string. Keywords together with their categories are shown in Table 3.

Table 3 Categorized keywords in elaborating the search string

Category	Keywords
C1	(non–functional requirements OR nonfunctional requirements OR non functional requirements OR quality attributes OR nonfunctional software requirements OR "quality characteristics" OR "quality factors" OR "quality requirements" OR "non-functional requirements" OR "non functional requirements")
C2	Software
C3	(elicitation OR selection OR select)
C4	(prioritization OR prioritizing OR prioritize OR prioritisation OR prioritising OR prioritise)
C5	(Specification OR specify OR specifying OR modelling)
C6	(metrics OR measurement OR measures)
C7	(validate OR validation OR validating OR test OR testable OR testability)

The search string used in this research is developed from Table 3 and is shown in Table 4.

Table 4 Search string formulation

Search string = Population AND Intervention AND Outcomes
Population (non–functional requirements OR nonfunctional requirements OR non functional requirements OR quality attributes OR nonfunctional software requirements OR "quality characteristics" OR "quality factors" OR "quality requirements" OR "non-functional requirements" OR "non functional requirements")
Intervention Software
Outcomes ((elicitation OR selection OR select) OR (prioritization OR prioritizing OR prioritize OR prioritisation OR prioritising OR prioritise) OR (Specification OR specify OR specifying OR modelling) OR (metrics OR measurement OR measures) OR (validate OR validation OR validating OR test OR testable OR testability))

4.5 Search venues

The following digital libraries are selected as search venues: ACM Digital Library, IEEE Explore, SpringerLink and ScienceLink. The libraries SpringerLink and ScienceDirect do not facilitate the use of complex search strings and we therefore decided to use Inspec/Compendex due to its extensive support for indexing references across several sources. SpringerLink and ScienceDirect were searched manually. This search was done by applying the search string to the title and abstract only. The reason for not applying the search string on the full text in search venues is that such searches yield too many irrelevant results [87].

4.6 Inclusion/Exclusion criteria and study quality assessment

As explained earlier, we have combined inclusion and exclusion criteria and study quality assessment in the review. The inclusion and exclusion criteria are used to identify studies relevant to the research questions. A delimitation factor is employed to identify studies that report evaluation, i.e. empirical findings. The delimitation factor for the review is selection of only empirical studies. The delimitation factor for the review is selection of only empirical studies. Evaluations can be on any scale, i.e. evaluations can be done on toy examples developed specifically for evaluation, down-scaled example or real example from industry (inspired by [49]). The following are the criteria for including a study.

- Articles published between 2000 and 2010 (including 2010)
- Articles only in English
- Articles should be available in full text

The following questions are used to select a paper for inclusion:

- Is the study relevant to software development for software-intensive products?
- Are the study goal/theme clearly stated?
- Is the study solely on functional requirements?
- Does the article present empirical evaluations?

If any one of the above questions is answered by a NO, the article/study is not included in the review. For example, to answer **RQ2 – "What is the current state of evaluation of RE for QR?"** and identify relevant primary studies, we use the answers to the question in the fourth bullet above. To identify the strength of evaluation and the amount of decision support to practitioner (RQ2.1 and RQ3), empirical papers are evaluated in this review. Pure research papers presenting new technologies are excluded as they do not present possible benefits and limitations. Furthermore, articles without a clear aim/goal cannot be used as a basis of technology adoption, such articles are excluded. The purpose of this review is to identify technologies relevant to QR. Technologies for functional requirements have been a major part in the study by Ivarsson [49] and studies focusing solely on functional requirements are excluded.

We have excluded editorials, newspaper editions, comments and expert review included in the review. The reason for this is to include only peer-reviewed research papers. Inclusion and exclusion criteria are applied on title and abstract. Runeson [88] provides a quality assessment checklist for case studies. The Dybå study [87] uses 11 criteria to assess the quality of studies. These quality criteria are identified and mapped onto a set of properties for data extraction: context described, study design and validity described. The criteria are adapted from [49] as this study [49] involves finding decision support for applicability of requirements engineering technologies published in RE journal based on research rigidity. The values and quality score associated with these properties are given in Table 5.

Table 5 Properties related to quality assessment and associated quality score

Property	Values	Quality score
Context described	Strong/medium/weak	Strong = 2, Medium = 1, Weak = 0
Study design	Strong/medium/weak	Strong = 2, Medium = 1, Weak = 0
Validity described	Strong/medium/weak	Strong = 2, Medium = 1, Weak = 0

4.7 Data extraction

To perform a review of the obtained primary studies, data from these are extracted and inserted into data extraction form. The data extraction form is shown in Table 6. This form explains the data extracted, and gives a brief description and mapping of the data to the research questions.

Table 6 Data extraction form description (adapted from [49, 87]

Unique primary study identifier				
Title of the study Goal/theme of the study Type of publication Year of publication				
#	Property	Values	Description	Mapping to RQ
1	Research method	As given in the study	Extracts the research method employed in the study. Examples of research methods are case study, experiment and survey	RQ2, RQ3
2	Context described	Strong/medium/ weak	Specifies to what extent context of the study is described	RQ3
3	Study design	Strong/medium/ weak	Specifies to what extent design of the study is described	RQ3
4	Validity discussed	Strong/medium/ weak	Specifies to what extent validity of the study is discussed	RQ3
5	Research context	Industry/academi a	Specifies the context in which the study took place.	RQ2, RQ3
6	Subjects	Practitioners/rese archer	Specifies the subjects involved in the empirical study	RQ2, RQ3
7	Scale of empirical evaluation	Industrial–real application/down -scaled real	Specifies the scale of empirical evaluation	RQ2, RQ3

		example/toy example		
8	Process area(s) addressed	As given in the study	To extract the process area described (process areas are elicitation, prioritization, specification, metrics, and testing)	RQ1, RQ3
9	QR addressed	As given in the study	To extract the type of QR used in the empirical study	RQ1, RQ3
10	Product domain/type of feature	As given in the study	To extract the type of application (and domain) used in the study	RQ1, RQ3
11	QR technology	As given in the study	To extract technologies addressing the process area addressed	RQ1, RQ3

Eleven properties were extracted from the primary studies. As shown in Table 6, these properties are mapped onto research questions and are allotted values as shown in Table 6, third column. These values are extracted from the primary studies. The process of assigning values to the extracted data is further explained below.

Properties 8-11 are extracted to answer RQ1. Property 8 shows the process area(s). A study may address more than one process area. Therefore, it is possible to mark this property more than once. Property 9 is the type of QR, which is used for the empirical evaluation. Property 10 is used to identify the type of software product used in the empirical evaluations. The motivation for extracting this property is to identify on what scale a technology is suitable for adoption. Property 11 is used to extract the technology itself. A technology can be a model, technique or a framework [75]. All the presented methods, techniques etc. will be referred as technologies in this thesis. Properties 8-11 answer RQ1and provide a database in relation to the QR process areas addressed in this thesis.

At the start of the review process, the author performed data extraction from a random sample of 10 papers and found certain difficulties in extracting the rest of seven properties (property 1-7). For example, many papers did not have a clear description of the research method used in the empirical evaluation. Similarly a brief description of the subjects of the empirical evaluation and how they are identified are not described in many of the papers. To have a more focussed way of evaluating the identified primary studies and the research methods used they used, we tried to obtain an understanding of the research methods indirectly. In some papers, the research methods are not directly described and in some papers the evaluations used an ambiguous terminology. Thus, to extract property 1, the following understanding of research methods is used to extraction this property:

- **Case study** if this is described as an empirical evaluation, or if the study uses an example or a scenario and performs the evaluation without specifying and having stated the goal of the evaluation [73]
- **Experiment** if this is mentioned and the study describes an experiment [73]

- **Survey** if this is mentioned and the study describes its procedure - irrespective of the amount of detail - or uses questionnaires or interviews to collect data from an identified population [84]
- **Lessons learned** if the study reports lessons learned, or describes industrial experiences after deploying a particularl method. The motivation for using this category is to gain insight into the practical use and resulting evidence of a particular method
- **Action research** if there is a section that explicitly describes the use of this method
- **Not stated** if the research method is not stated or does not fall into the above described categories.

To answer RQ2, properties 5-7 are extracted and used together with property 1. From a technology transfer perspective, these properties convey the evidence for the effect of using a method [49, 75]. The three properties describe the environments and circumstances under which a method is adopted and evaluated and underlines the derivation of the evidence [49].

Property 5 - Research context - captures the context in which the empirical evaluation was done. Studies are classified into industrial and academic studies. They are classified as industrial if the empirical evaluation is done in an industrial setting which is explicitly mentioned or if the empirical evaluation was performed in collaboration with industry. In all other cases, the property is given the value “academia”, i.e. studies performed in an academic setting such as a laboratory experiment or in a setting for which the research context is not described.

Property 6 describes the subjects involved in the empirical evaluation. A distinction is made between practitioners and researcher. For example, if the evaluation is performed with practitioners as subjects and this is clearly stated, the property is assigned the value ‘practitioner’. If the study is performed by the researcher himself using an example or if the study is performed using students as study subjects, the property is assigned the value ‘researcher’. If the study does not explicitly mention who the subjects of the evaluation are, property 6 is assigned the value ‘researcher’.

Property 7 is called Scale of evaluation. The motivation for extracting this property is to identify the scale of application utilized to empirically evaluate a technology (inspired by [49]). For example, a study performed using a real industrial application can offer better evidence for practitioners wishing to adapt a technology than a study performed using a toy example invented for the evaluation. The values associated with property 7 are toy example, down-scaled real example and industrial-real application. Toy examples are constructed as examples, often invented for evaluating the application while down-scale applications are based on real example but are modified for the purpose of the study.

The properties 2-4 are extracted to identify the strength of evidence for the methods. Based on the values extracted for these properties, a quality score is calculated according to the rule given in Table 5. The associated values are extracted based on the extent to which these values are described in the paper.

Property 2 is scored on a three level scale. The property is assigned a value *weak* if the context of the research (i.e. concerning the evaluation part of the study) is not mentioned at all or the context of the evaluation cannot be extracted. *Medium* score is assigned when the context is described but not in detail. *Strong* score is assigned if the research context is clearly described, for example type of application used, and description of the development effort.

Property 3 is related to the research design used for the empirical evaluation. *Strong* score is assigned if the study describes the research design clearly - including the choice of research method, data collection procedures and procedure for identification of subjects in the evaluation. *Medium* score is assigned if the research design is not described in detail. For example, if the study does not explicitly explain how the subjects in the empirical evaluation are identifies and given tasks, *medium* score is assigned. *Weak* score is assigned if the research design is not described at all.

Property 4 concerns the validity of the empirical evaluation and how the results are presented. *Weak* score is assigned if the validity of the study is not presented at all. *Medium* score is assigned if the validity of the study is presented but not in detail, or a description of the limitation of the study is given. *Strong* score is assigned if the validity of the study, various threats involved in the study, limitations of the study design, procedure and the results are described.

4.8 Conducting the review

Conducting the review involves searching the identified digital libraries and databases using the search string. This was done manually for each of the databases. It resulted in a total of 2380 papers. The author decided to use a trial version of a reference management system called *Endnote*. After removing duplicates and applying inclusion and exclusion criteria to title and abstract, a total of 232 papers were left for full text review. Of these 232 papers 186 papers were excluded as they did not present empirical evaluations in relation to QR. In total, 46 papers were left for review and data extraction. Table 7 shows the primary studies identified in the review.

Table 7 Primary studies

#	Study theme/goal	Year	Reference
1	To determine if and how the recommended changes from the usability test were implemented, to identify other changes made to the feature being designed and the source of those other changes	2000	[1]
2	To identify practical software metrics for intranet applications	2001	[2]
3	Presenting and validating a framework for integrating non-	2001	[3]

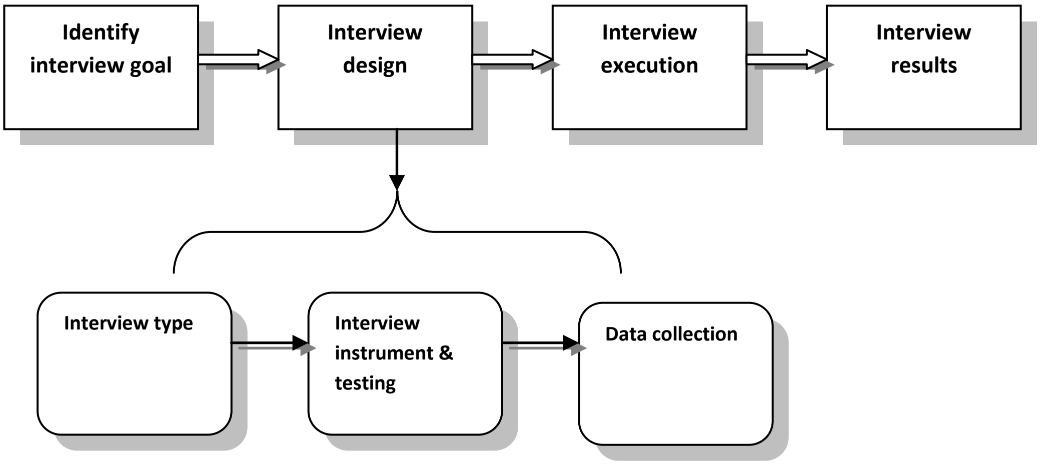
	functional requirements into the ER and OO models		
4	Elicitation and evaluation of quality requirements using the ATAM approach	2001	[4]
5	SHIRA method to explore the meaning of abstract product qualities in a specific context from users’ perspective	2001	[5]
6	Application of the WinWin paradigm to identify and resolve conflicts in a series of real-client, student-developer digital library projects	2001	[6]
7	A model to identify and specify quality attributes that crosscut requirements including their systematic integration into the functional description at the requirements stage	2002	[7]
8	Investigating how architects consider quality requirements and what are the most influential quality requirements types	2010	[8]
9	A process to elicit NFRs, analyze their interdependencies, and trace them to functional conceptual models	2004	[9]
10	A tool to validate non-functional system requirements, such as system reliability	2004	[10]
11	Requirements Engineering Framework (REF) that allows early adoption of system functionalities and quality attributes	2004	[11]
12	Identifying, extracting and generating quality migrant object oriented code that satisfies non-functional requirements	2004	[12]
13	Application of a systematic, experience-based method to elicit, document and analyze non-functional requirements with a objective to achieve a minimal set of measurable and traceable non-functional requirements	2005	[13]
14	A method for eliciting and prioritizing security requirements (SQUARE)	2005	[14]
15	Presents goal-object pattern framework to capture and model functional and non-functional using UML and goal-oriented method	2006	[15]
16	A practical framework for eliciting and modeling dependability requirements	2006	[16]
17	An evolutionary model for performance requirements specification and corresponding validation	2006	[17]
18	Describes use/role of usability testing in agile projects	2006	[18]
19	An approach to the identification and inclusion of non-functional aspects of a business process in modeling business improvement	2007	[19]
20	Testing approach of component security (TACS) based on dynamic monitoring and detecting algorithm CSVD	2007	[20]
21	An XML-based software non-functional requirements modeling method	2007	[21]
22	Requirements elicitation tool (ElitiO)to capture precise non-functional requirements specifications during elicitation interviews	2007	[22]
23	Evaluation of aspect-oriented techniques in testing non-functional requirements of an industrial system	2007	[23]
24	Presents QRF (Quality requirements of a software family) method focusing on defining, representing quality requirements and transforming to architectural models	2007	[24]
25	Presents a model to incorporate quality as a dimension used in prioritization of functional requirements	2007	[25]

26	Discussion of security objectives and security requirements	2007	[26]
27	Aspectual support for specifying requirements for software product lines	2007	[27]
28	Process framework for customizing software quality	2007	[28]
29	A Tool for measuring user experience and usability	2007	[29]
30	UML profile for modeling non-functional requirements in a generic way	2007	[30]
31	Goal-oriented requirements engineering (GORE) for eliciting security requirements	2008	[31]
32	To understand how the availability and precision of performance requirements , “Not a Problem” (NaP) defect report, and PREM model	2008	[32]
33	A method based on software quality spectrum embedded in a software engineering artifact	2008	[33]
34	Presents SPUR – for modeling cross-functional attributes of software	2008	[34]
35	To support roadmapping of quality requirements	2008	[35]
36	A framework for software safety based on McCall’s software quality model	2008	[36]
37	To measure software system dependability in architecture design phase	2009	[37]
38	A prototype and usability test of a Near Field Communication (NFC) –based ticketing application	2009	[38]
39	A security testing approach that derives test cases from design level artifacts	2009	[39]
40	A methodology based on the extension of Product Line UML-based Software Engineering (PLUS) techniques to model NFR	2009	[40]
41	To identify the unique challenges associated with the selection, trade-offs and management of QR	2009	[41]
42	A lightweight group method that helps relevant stakeholders to elicit, prioritize and elaborate the quality goals of a software product	2009	[42]
43	An approach to software performance testing	2000	[43]
44	To investigate which qualities are considered the most expensive to obtain, as well as which are the most wanted	2001	[44]
45	To support intuitive and systematic identification of quality requirements	2008	[45]
46	SecReq method for security requirements engineering	2010	[46]

CHAPTER 5
Industrial interviews

In addition to a SLR, an empirical study was performed as part of the thesis. In the empirical part we interviewed people from the software industry. Interview is one of several techniques used for collecting qualitative data [88]. According to [88], interviews should be conducted to collect data based on observations, memories and opinions of interviewee in a particular setting [88]. The purpose of our industrial interviews in relation to the thesis is to study how the QR technologies are used in industrial. The interviews were conducted using semi-structured interviews [88] and the interview process is described in Figure 4.

Figure 4 Industrial interview process



5.1 Interview goal

The purpose of conducting industrial interviews is to supplement SLR results by obtaining insights into the industrial practices in adopting QR technologies to handle QR. To this end, industrial interviews aim at identifying most opted/selected QR for implementation, importance of QR in software projects, and technologies used for the incorporation of measurable and testable QR.

5.2 Interview design

Interviews are one of the commonly used methods for collecting qualitative data [88]. The design of the interviews should be aligned with the identified goal and purpose of the interview. First we need to develop the interview guide - identifying how the interviews will be conducted and how the data will be collected and analyzed. In addition we need to decide on Interview type, Interview instrumentation and testing and Data collection.

5.2.1 Interview type

There are three types of interviews - *structured interviews, semi-structured interviews* and *unstructured interviews* [88]. In a semi-structured interview questions are open ended, allowing broad discussions on the topics, thus offering flexibility [88] and the opportunity for follow-up questions. In this thesis, we have used semi-structured interviews. Closed questions should only be used for questions that are objective in nature. Open-ended questions are used to elicit unexpected information during the interview. The low availability of the experts from industry made us use semi-structured interviews as a flexible way of eliciting information that fitted our goals. The section *interview execution* will describe how the interviews were conducted and the flexibility that was needed.

5.2.2 Interview instrument and testing

A questionnaire was used to elicit information on the identified interview questions. The questionnaire was divided into two parts: personal and goal specific. The interview questionnaire and a brief description of the goals are shown in Appendix A. When the questionnaire has been designed it is important to validate that the questionnaire will meet the identified goals. For this reason, questionnaires need to be tested before conducting interviews in a real setting. The questionnaire was verified by the supervisor who provided comments and feedback. This resulted in easy-to-understand questions and improved the wording of the questions. Furthermore, the questionnaire was tested with two students who had prior work experience from industry. The students gave comments and tips on how to conduct an interview in a real setting. The two students found the questions understandable and answerable. In addition, the author had prior experience in conducting interviews. All of this was input to the design and testing of the questionnaires.

5.2.3 Data collection

During the interviews, data will be collected by listening to the interviewees and taking notes and recording answers. Permission will be granted from the participants for recording interviews. Once the answers are recorded and written down, they will be analyzed. That is, the answers are interpreted in accordance with the research questions of SLR.

5.3 Interview execution

The first step in executing the interviews is to identify participants, i.e. experts from industry. The author identified two participants one from Karlskrona, Sweden and one from Athens, Greece. There were no predefined criteria when selecting the interview participants from companies developing software-intensive products. The first participant was identified via a common friend while the second participant was identified through a business-oriented social networking site – Linkedin – [89]. Participants were briefed orally about the purpose and goals of the interview. The author could not conduct the interview with the first participant because of management issues at the company. The second participant (later referred as Participant A) agreed to participate. Two more participants (hereafter referred as Participant B and Participant C) were identified with the help from the supervisor. Participant A is situated in Athens, Greece while Participants B and C are situated in Trondheim, Norway. All three final participants preferred a telephonic interview before the interview questionnaire was filled in. Firstly, due to issues of availability questionnaires were answered offline and the answers were returned to the author through e-mail. At the second stage, after analyzing the answers, a few more questions were added to get more insight. This was done as per the convenience and as suggested by the participants. Questionnaire was sent to participant C also, but this participant did not provide further answers to the questionnaire as the questions found unanswerable from the perspective of QR practices at the company (See also Chapter 7). Participant A and B gave the permission to use their names in the thesis report. For the reasons of confidentiality Participant C wished to remain as anonymous. The interview questions with description/motivation are given in Appendix A. The participants' answers are given in Appendix B - D.

CHAPTER 6
Systematic literature review results

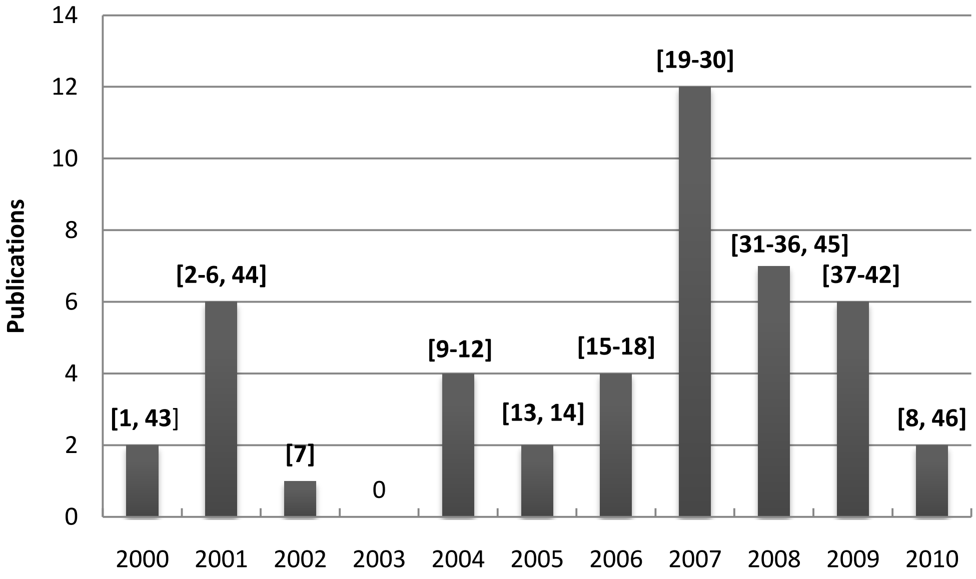
6.1 Systematic literature review (SLR) results

A total of 46 papers were reviewed. The results of the SLR are divided into sub-sections. Prior to describing the results from the SLR and the answers to the research questions, we will give an overview of the studies.

6.1.1 Publication year

A general overview of studies published per year is given in Figure 5. It should be noted that a majority of the studies - 27 publications (59%) were published in the last four years (2007-10). A sudden increase in the publications is observed in the year 2007 with 12 publications (26%). A possible reason for this is an increased interest in and attention to QR research. As the majority of the studies are from the recent years, studies obtained in the review can be of substantial importance in terms of research directions.

Figure 5 Distribution of studies according to year of publication. The references are included in square brackets.



6.1.2 Research methods

Of the 46 reviewed studies, a total of 35 (76%) studies utilized case studies as research method. The number of experiments, surveys and studies that reported “lessons learned” is 3, 4 and 4 respectively. Case studies generally offer a high level of realism when conducted in an industrial setting [49]. It should be noted that several of the surveyed articles did not have a section for research methodology, neither was the name of the research method used mentioned. In such cases articles with clear cut research goals are classified as case studies. Extracting the research method is done using the rules given in Chapter 4.7. Inclusion and exclusion criteria are formed in such a way that they favour the process of identifying research methods and extracting data from articles. This is because articles with clearly specified research goals/theme are included in the review. Articles that did not have clearly defined goal/theme were not included in review.

Table 8 shows the distribution of research methods in the reviewed studies.

Table 8 Distribution of research methods across studies

Research method	Number of studies	Percentage
Case studies	35	76
Experiments	3	7
Surveys	4	8
Lessons learned	4	8

6.1.3 Research context

Studies are classified as either Industry or Academia based on the study setting for the empirical evaluations. Evaluations (giving positive results) performed in industrial setting increase the possibility of technology transfer to a different industry setting or context [49]. Table 9 shows the distribution of studies based on the property Research context.

Table 9 Classification of studies based on Research context

Research context	Number of studies	Percentage
Industry	24	52
Academia	22	48

6.1.4 Scale of empirical evaluation

The scale of the applications used in the empirical evaluation affects the evidence produced by the evaluations [49]. Studies are classified based on the applications scale in the evaluations. This is extracted by observing the applications used in the evaluations. Three types of applications scale are used in the empirical evaluations: toy examples, down-scaled real example, and industrial-real applications. Table 10 shows the distributions of applications scale utilized in the reported studies.

Table 10 Applications scale utilized

Scale of empirical evaluation	Number of studies	Percentage
Industrial-real application	25	54
Down-scaled real example	11	24
Toy example	10	22

6.1.5 Subjects

Subjects in an empirical study will strongly influence the kind of evidence the study can report. Subjects are classified as either practitioners or researchers. Table 11 shows the number of studies using practitioners and researchers as subjects in the empirical evaluations.

Table 11 Subjects in the studies reported

Subjects	Number of studies	Percentage
Practitioners	21	46
Researchers	25	54

As shown in Table 11, more than 25 studies (more than 54%) involved researchers as subjects in empirical evaluations. Studies that involved students as subjects are counted as belonging to the researcher category. According to the technology transfer model [78], adoption of a certain technology first involves piloting a technology in industry, i.e. in a real setting involving practitioners. For this reason, the property “subjects” has two classifications: practitioners and non-practitioners. A possible reason for such a high number of studies involving researchers as subjects in the empirical evaluations is that it is difficult to get permissions to pilot a technology in industry. In addition, the time and cost needed when introducing new technologies in industry is high. This problem was also observed by Ivarsson [49]. Another possible reason for high number of studies involving researchers as subjects is the need to validate a technology in an academic setting before trying to convince practitioners of the benefits.

21 (46%) studies involved practitioners as subjects in the evaluations. However, it should be noted that majority of the studies did not give any information on the subjects or just have a brief introduction of who the subjects were. The property Subjects often had to be inferred from the studies by carefully going through the research methods sections. In cases where the researchers themselves were employed in industry, the property “subjects” was marked as practitioners. Also, some studies were a result of close cooperation with industry involving both researchers and practitioners as subjects. In such cases the property was marked as practitioners.

6.1.6 Quality scores of studies

Properties 2-4 are used to give a quality score to each of the studies reported in the review. Quality scores are generated using the rules in Table 5. In order to get an over-all score, we used a combination of the properties 2 - 4 (context described, study design and validity discussed). This is done by adding the individual scores of the properties for each study. Table 12 gives an overview of the quality scores of the reported studies.

Table 12 Overview of quality scores of studies

Quality score	0	1	2	3	4	5	6	Total
Number of studies	4	3	5	11	12	5	6	46

Of the total 46 studies reviewed, only 6 studies (13%) scored 6, i.e. full score. 10 % of the studies scored 5. It can be observed that general quality of the reviewed studies is low with a mean of 3.7, mode 4 and median 3.5. An overview of quality scores achieved by individual properties (properties 2-4) is given in Table 13.

Table 13 Overview of quality scores of data extraction properties contributing to study quality

Property	0 (weak)	1 (medium)	2 (strong)	
Context described	5	11	30	mean = 1.5, mode = 2, median = 2
Study design	6	19	21	mean = 1.3, mode = 2, median = 1
Validity discussed	29	11	6	mean = 0.5, mode = 0, median = 0

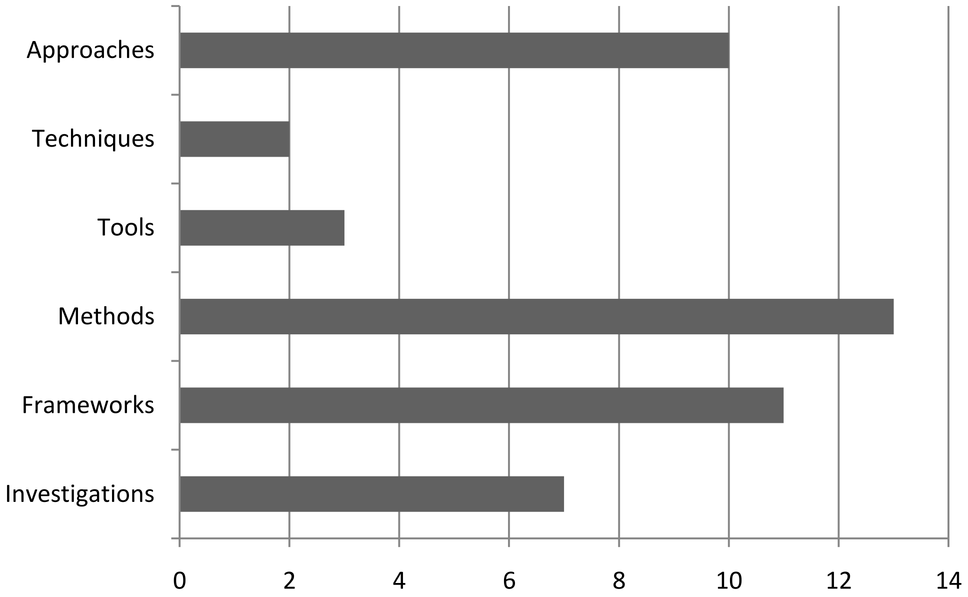
6.1.7 Classification of technologies

Studies are classified based on the presented technologies. These classifications are: methods, techniques, software engineering tools, framework/model/process, and approach/prototype. Several studies did not present technology per se. Instead they used empirical methods to investigate a phenomenon. Such studies are classified as investigations. Figure 6 gives an overview of the studies and technologies presented. Apart from investigations the presented technologies are divided into five categories: Approaches/Processes, Techniques, Tools, Methods and Frameworks/Models. The following definitions are used for these classifications:

- **Processes** if the technology presented a sequence of steps, or a procedure/approach or a framework to perform a certain task(s) for a given purpose(s), or if it is explicitly mentioned in the article.
- **Techniques** if it is explicitly mentioned in the article.

- **Tools** if the articles describe software artefacts that aid in software development activities.
- **Methods** if the article explicitly mentions a “Method”, otherwise the technology is treated as sequence of steps and thereby classified as a Process.
- **Models/Frameworks** if the article explicitly mentions a Models or Framework, or defines a set of activities and that requires tailoring to a given set of needs.

Figure 6 General overview of technologies presented



The classifications are loosely defined. This was done after a preliminary scanning of the primary studies. Many of the primary studies explicitly mentioned one of the above classifications.

6.1.8 Process areas discussed

The thesis reviewed articles that presented evaluated technology focusing on how to handle QR in relation to elicitation, prioritization, metrics, specification and testing. Of the 46 studies, elicitation was addressed in 18 studies, prioritization in 11 studies, 13 addressed metrics. 15 studies addressed specification and testing.

Table 14 General overview of process areas addressed in the reported studies

Process area addressed	Research context = industry	Research context = academia	Total studies
Elicitation	[3], [4], [5], [9], [13], [17], [35], [41], [42], [45], [46]	[6], [11], [14], [16], [22], [24], [36]	18
Prioritization	[2], [8], [13], [25], [28], [34], [41], [42], [44]	[14]	11

Metrics	[2], [23], [25], [28], [29], [32], [35]	[12], [16], [22], [24], [30], [37]	13
Specification	[17], [23], [34], [35], [42], [46]	[7], [15], [16], [19], [20], [21], [22], [24], [27], [31], [40]	17
Testing	[1], [17], [18], [23], [26], [28], [29], [43]	[10], [19], [20], [33], [37], [38], [39]	15

A general overview of which studies discuss which process area and have which research context is given in Table 14. Note that several studies address more than one process area.

6.2 *Views on QR selection (RQ 1.1)*

The purpose of the research question RQ1.1 - "What are the most selected/opted QR?" was to identify views and attitudes towards incorporating QR into software. Overall, 6 studies reported findings in relation to RQ1.1. Table 15 gives an overview of the relevant findings - the research context or setting of the study, the research method used and the subjects of the study.

Table 15 General overview of studies reporting views and attitudes towards QR

Study description	Application (name/domain) reported	Reference
• **The six software quality characteristics and 32 quality sub-characteristics of the Extended ISO model are used as a basis for identifying key quality characteristics for intranet applications.**	Intranet applications	[2]
• **A web survey of software architects used to identify the most influential types of QR.**	N/A	[8]
• **A survey of software engineers in a company to identify important characteristics of two different products based on ranks using ISO 9126-2 quality model.**	N/A	[28]
• **Interviews of five project leaders and five product managers from five companies identifying the most important quality aspects**	Embedded systems	[41]
• **Survey of software architects and system designers to identify the views on implementing QR for software platforms**	Embedded systems	[44]

Views and attitudes of practitioners towards QR have been observed in five studies (Table 15). Study [8] reports from a survey conducted with the purpose of identifying how architects consider QR and what the most influential types of QR are in their daily work. The types of QR that are considered most important are efficiency, maintainability, reusability, reliability and usability. Only 57% of the survey respondents (of 60 responses) use QR to make architectural and technical decisions. Although the survey sheds some light on the most important QR as conceived by architects, software products and domains for which the identified QR are perceived as the most important are not mentioned.

Leung [2] performed a survey to identify the key QR for intranet applications. The extended ISO quality model [89] was used as a survey instrument for deriving quality characteristics and sub-characteristics. The six quality characteristics and the thirty two sub-characteristics were ranked. The results of the survey show that the three most important quality characteristics are reliability, functionality and efficiency. Five sub-characteristics were found to be important for intranet applications - namely availability, accuracy, security, suitability and time behaviour, with availability ranked first and suitability last. The mapping of sub-characteristics to their corresponding quality characteristics shows that availability is considered the most important reliability attributes, time behaviour the most important efficiency attribute while accuracy, security and suitability are considered the most important functionality attributes.

Sibisi and Waveren proposed a framework to customize software quality models to the product needs [28]. They applied the framework in a real working environment and validated it. To customize ISO/IEC 9126 to the needs of the company UEC Technologies, they conducted a survey with ten software engineers to identify the important quality characteristics for two products (details of the products are not mentioned for the reasons of confidentiality) [28]. Six quality characteristics were chosen for the study - functionality, maintainability, reliability, portability, usability, and efficiency. The order of importance for the quality characteristics were observed for both products. The order of importance of quality characteristics for the products is the following: Product X - functionality, efficiency, reliability, usability, portability and maintainability, Product Y - functionality, reliability, usability, efficiency, maintainability and portability. The level of importance for the same quality characteristic is not same for the products X and Y. For example, efficiency is rated higher for, while maintainability is considered to be less important for the product X. A framework suggested in the study [28] can be used as a guide when customizing the quality model.

The studies [41, 44] report survey results from practitioners developing embedded systems. Svensson et al. [41] explore the views of project leaders and product managers on types of QR. Usability and performance were considered to be the two most important QR for implementation of control and telecom systems. The priorities of project leaders and product mangers in selecting types of QR differed. The priorities of project leaders were found to be usability followed by performance, compliance and flexibility, while product managers considered performance as the most important type of QR followed by usability and security.

Johansson et al. [44] conducted a survey in two organizations (B and C) to identify the views of software architects and system designers on QR. The organizations are large and develop embedded systems for telecom industry. In organization B, architects and designers consider reliability to be the most important quality aspect. The order of priority of quality aspects in organization B is maintainability, reusability, functionality, usability and functionality. The observed order of priorities in the case of organization C was reliability, functionality, maintainability, reusability, usability and efficiency. The differences in the priorities can be found from the survey results. Although there are differences in the priorities, functionality, maintainability and reliability were considered to be the most important quality aspects to include in the products by architects and designers.

6.3 *Technologies and themes addressed for QR in relation to elicitation, prioritization, specification, metrics and testing (RQ 1.2)*

The purpose of this RQ was to identify the technologies presented for QR in relation to the five process areas. As observed in Table 14, several studies address more than one process area. For example, the study by [16] addresses three process areas: elicitation, specification and metrics. To give a better analysis of the process areas addressed and technologies presented, the author identified 16 combinations of process areas addressed among the 46 studies reviewed. Table 16 gives an overview of the process areas and the 6 identified categories of the technologies presented.

Table 16 Overview of technologies evaluated to address process areas

ID	Process areas	Approaches (A)	Techniques (B)	Tools (C)	Methods (D)	Frameworks/ models (E)	Investigations (F)
1	Elicitation	[4], [31]	-	-	[5], [6], [45]	[3], [9], [11], [36]	-
2	Prioritization	-	-	-	[34]	-	[8], [44]
3	Specification	-	[27]	-	[21], [40]	[7], [15]	-
4	Metrics	[12]	-	-	-	-	[32]
5	Testing	[18], [38], [39], [43]	-	[10]	[33]	-	[1], [26]
6	Elicitation & prioritization	-	-	-	[13], [14]	-	[41]
7	Elicitation &specification	-	-	-	[46]	-	-
8	Prioritization & metrics	-	-	-	-	[25]	[2]

9	Metrics & specification	[30]	-	-	-	-	-
10	Metrics & testing	-	-	[29]	[37]	-	-
11	Specification & testing	[19], [20]	-	-	-	-	-
12	Elicitation, prioritization & specification	-	-	-	[42]	-	-
13	Elicitation, specification & metrics	-	-	[22]	[24]	[16], [35]	-
14	Elicitation, specification & metrics	-	-	-	-	[17]	-
15	Metrics, specification, & testing	-	[23]	-	-	-	-
16	Metrics, prioritization & testing	-	-	-	-	[28]	-

Analysis: A plethora of technologies has been evaluated in relation to the QR process areas. For example, the review identified 12 methods that were empirically evaluated to address the 16 combinations of process areas, (see Table 16). From Table 16 we observe that there is considerably few tools (3 studies out of 46) evaluated to address QR. To give a better look at the technologies and process areas, the properties 5, 7, 9 and 10 are combined for each process area to identify the extent to which each process area has been addressed. The results are presented in Tables 17-33. The technologies presented are marked by letters corresponding to their representations in Table 16. The following representations are used: A: Approaches, B: Techniques, C: Tools, D: Frameworks/Models, E: Investigations.

Table 17 Objective analysis of technologies addressing elicitation

Study	Research context	Scale of empirical evaluation	QR addressed	Product domain/type of feature	Technology presented	
[4]	Industry	Industrial-real	General	Factory process controlling system	ATAM	A
[31]	Academia	Down-scaled	Scalability	Enterprise system	GORE	
[5]	Industry	Down-scaled	General	Home automation system (HAS)	SHIRA	D
[6]	Academia	Down-scaled	Dependability, usability, reusability, performance	student projects: digital library systems	WinWin	

[45]	Industry	Industrial-real	General	Data base system	MOQARE	
[3]	Industry	Industrial-real	General	Information system for a clinical analysis laboratory	Framework to integrate QR into ER and OO models	E
[9]	Industry	Industrial-real	General	Information system for clinical analysis laboratory	Framework to elicit QR and trace them to functional requirements	E
[11]	Academia	Down-scaled	General	Aircraft simulation system	REF	
[36]	Academia	Down-scaled	Software safety	Road traffic control system	Framework for software safety	

Table 18 Objective analysis of technologies addressing prioritization

Study	Research context	Scale of empirical evaluation	QR addressed	Product domain/type of feature	Technology presented	
[34]	Industry	Industrial-real	General	Vehicle consumer services interface system	SPUR	D
[8]	Industry	Industrial-real	General	N/A	-	F
[44]	Industry	Industrial-real	Efficiency, functionality, reliability, usability, reusability, maintainability	Embedded system	-	

Table 19 Objective analysis of technologies addressing specification

Study	Research context	Scale of empirical evaluation	QR addressed	Product domain/type of feature	Technology presented	
[27]	Academia	Toy example	General	Health watcher system	Aspects based	B
[21]	Academia	Toy example	Performance	N/A	XML-based	D
[40]	Academia	Toy example	Usability, security, performance	Web-base information system	Extension of PLUS	
[7]	Academia	Down-scaled	Response time, security	Toll collection system	Model to specify quality attributes	E
[15]	Academia	Toy example	Scalability, security	Online bookstore application	Goal-object pattern framework	

Table 19 (continued)

Table 20 Objective analysis of technologies addressing metrics

Study	Research context	Scale of empirical evaluation	QR addressed	Product domain/type of feature	Technology presented	
[12]	Academia	Toy example	General	Open source procedural systems	Strategy to incorporate QR in software migration process	A
[32]	Industry	Industrial-real	Performance	Real-time embedded control system	-	F

Table 21 Objective analysis of technologies addressing testing

Study	Research context	Scale of empirical evaluation	QR addressed	Product domain/type of feature	Technology presented	
[38]	Academia	Down-scaled	Usability	Ticketing system	Prototype and usability test	A
[39]	Academia	Down-scaled	Security	Shopping cart application	Security testing approach	
[43]	Industry	Industrial-real	Performance	Client/server transaction processing application	Software performance testing approach	A
[18]	Industry	Industrial-real	Usability	Web-service application	"Wizard of Oz" using a paper prototype approach	
[10]	Academia	Down-scaled	Reliability	Military frigate combat system	Tool to validate system QR	C
[33]	Academia	Industrial-real	General	Web-browser	Quality spectrum based method	D
[1]	Industry	Industrial-real	Usability	Automated file synchronization system	-	F
[26]	Industry	Industrial-real	Security (confidentiality, integrity, availability)	Telecommunication system	-	

Table 22 Objective analysis of technologies addressing elicitation & prioritization

Study	Research context	Scale of empirical evaluation	QR addressed	Product domain/type of feature	Technology presented	
[13]	Industry	Industrial-real	Security, Efficiency, Reliability	Web-based geographical information system (GIS)	Systematic, experience-based method	D
[14]	Academia	Industrial-real	Security	Asset management system (AMS)	SQUARE	
[41]	Industry	Industrial-real	General	Embedded system	-	F

Table 23 Objective analysis of technologies addressing elicitation & specification

Study	Research context	Scale of empirical evaluation	QR addressed	Product domain/type of feature	Technology presented	
[46]	Industry	Industrial-real	Security	Internet protocol television (IPTV)	SecReq	D

Table 24 Objective analysis of technologies addressing prioritization & metrics

Study	Research context	Scale of empirical evaluation	QR addressed	Product domain/type of feature	Technology presented	
[25]	Industry	Industrial-real	Interoperability, usability, security, reliability	Mobile handset	QUPER	E
[2]	Industry	Industrial-real	Efficiency, availability, accuracy, security, suitability	Intranet applications	-	F

Table 25 Objective analysis of technologies addressing metrics & specification

Study	Research context	Scale of empirical evaluation	QR addressed	Product domain/type of feature	Technology presented	
[30]	Academia	Toy example	Reliability, scalability, performance	Caching service example	UML based approach	A

Table 26 Objective analysis of technologies addressing metrics & testing

Study	Research context	Scale of empirical evaluation	QR addressed	Product domain/type of feature	Technology presented	
[29]	Industry	Industrial-real	Usability	Telecom - mobile	Tool to measure user experience	C
[37]	Academia	Toy example	efficiency, availability, accuracy, security, suitability	Video conference system	Quantified dependability analysis model framing method	D

Table 27 Objective analysis of technologies addressing specification & testing

Study	Research context	Scale of empirical evaluation	QR addressed	Product domain/type of feature	Technology presented	
[19]	Academia	Toy example	Quality information	Cancer registration system	NFR framework based approach	A
[20]	Academia	Toy example	Security	Email client	Testing approach of component security	

Table 28 Objective analysis of technologies addressing elicitation, prioritization & specification

Study	Research context	Scale of empirical evaluation	QR addressed	Product domain/type of feature	Technology presented	
[42]	Industry	Industrial-real	General	N/A	Lightweight method to elicit, analyze quality goals	D

Table 29 Objective analysis of technologies addressing elicitation, specification & metrics

Study	Research context	Scale of empirical evaluation	QR addressed	Product domain/type of feature	Technology presented	
[22]	Academia	Toy example	Efficiency, time behaviour	University web development	ElicitiO	C
[24]	Academia	Toy example	Performance Reliability	DiSep case example	QRF	D
[16]	Academia	Down-scaled	Dependability	Tactical separation assisted flight	Framework for eliciting and modeling	E

				environment (TSAFE)	dependability requirements
[35]	Industry	Industrial	Performance, utility	Mobile application	QUPER model

Table 30 Objective analysis of technologies addressing elicitation, specification & testing

Study	Research context	Scale of empirical evaluation	QR addressed	Product domain/type of feature	Technology presented
[17]	Industry	Industrial-real	Performance	Mission-critical software in retail industry	PREM

E

Table 31 Objective analysis of technologies addressing Metrics, specification, & testing

Study	Research context	Scale of empirical evaluation	QR addressed	Product domain/type of feature	Technology presented
[23]	Industry	Industrial-real	Performance (execution time, latency)	System for quality verification of mobile phones	Aspect oriented techniques for testing QR

B

Table 32 Objective analysis of technologies addressing metrics, prioritization & testing

Study	Research context	Scale of empirical evaluation	QR addressed	Product domain/type of feature	Technology presented
[28]	Industry	Industrial-real	Reliability, usability	Mission-critical software in retail industry	Framework to customize quality models

E

As mentioned earlier, a total of 16 combinations of process areas have been observed in the reported studies. 24 studies have been evaluated in an industrial context and 25 studies utilized industrial applications. A variety of applications have been observed. Embedded systems (mobile phones) have received more attention than other types of software products like business-critical systems. The review identified 23 studies that evaluated technologies in industrial context utilizing industrial applications [1, 3, 4, 8, 9, 13, 17, 18, 23, 25, 28, 29, 32, 34, 35, 41, 42, 43, 44, 45, 46]. Of these 23 studies 8 studies [1, 2, 8, 18, 26, 32, 41, 44] are primarily investigative in nature. That is, these studies were investigating or exploring a phenomenon – e.g. study [41] investigated how QR are handled in practice and various challenges involved in specifying measurable requirements. The QUPER model has been reported in two studies [25, 35]. The QUPER model is currently being used in industry to elaborate quality aspects of the product and defining quality levels that should be present in the product. A lightweight method [42] adapted QUPER model to elicit, analyze, and specify quality goals.

6.4 *Current state of evaluation of RE for QR (RQ2):*

The purpose of this research question was to assess the state of evaluation of technologies in relation to QR. The evaluation is done by assessing the realism of the studies. The realism score is found by combining the scores for research methods, research context, subjects involved in the evaluation and the scale of application used for the evaluation itself [49]. The idea of assessing the realism is inspired by [49]. Tables 8- 11 are combined and the result is shown below in Table 33. The combination is achieved by combining the properties Research method, Research context, Subjects, Scale of evaluations and the number of studies for each combination of these properties.

Table 33 Realism offered by the studies

Research method	Research context	Subjects	Scale of evaluation	# of studies	
Case study	Industry	Practitioners	Industrial-real	10	41%
Experiment	Industry	Practitioners	Industrial-real	1	
Survey	Industry	Practitioners	Industrial-real	4	
Lessons learned	Industry	Practitioners	Industrial-real	4	
Case study	Industry	Practitioners	Down-scaled	1	
Case study	Industry	Researchers	Industrial-real	4	
Case study	Academia	Researchers	Down-scaled	8	
Case study	Academia	Researchers	Industrial-real	2	
Experiment	Academia	Researchers	Down-scaled	2	
Case study	Academia	Researchers	Toy example	9	
Case study	Academia	Practitioners	Toy example	1	

Analysis: All in all 11 combinations can be found by combining research methods, research context, subjects involved in the evaluation and the scale of application used for the evaluation. As shown in the table above, 41% of the studies (19 out of 46) were conducted in industry and by practitioners or involving practitioners as subjects and using an industrial application. This result is encouraging in terms of evaluations offering realism. This also means that more evaluations need to be performed in realistic settings so that they can give a realistic picture, i.e. use of technologies in industry context and thereby offer decision support in technology adoption to practitioners.

As many as 60% of the studies do not offer the realism needed from a technology transfer perspective. Of the 35 case studies presented in the review, only 10 (28%) have a high degree of realism. Four studies reported “lessons learned” from industry experiences. Of the 25 studies performed by researchers, 6 studies utilized an industrial application. This means that the rest of the studies were performed either on down-scaled applications or on toy examples.

This result is not encouraging in the sense only six studies out of 25 (24%) performed by researchers have used an industrial application. This further implies that practitioners who would like to adopt technologies evaluated by researchers are limited by the number of studies and evaluations in realistic settings. Practitioners are limited by the scientific rigor found in the studies (See Table 12).

6.5 *Strength of evidence (RQ3):*

For technology transfer and adoption, strength of the evidence of the studies must be evaluated. The reason for this is that studies with strong evidence are good sources of information when building confidence to the information needed when adopting a method or technology in a particular industrial setting. Thus our review identifies the strength of evidence available by adopting the methodology followed in Ivarsson [49]. To identify the extent to which a study has been described, empirical evaluations with high degree of realism (See Table 33) are combined with the properties 2-4 (context described, study design, and validity discussed). To support technology transfer, evaluations should provide a realistic picture so that results produced in a realistic setting can be transferred to a new industrial setting. Table 34 gives an overview of studies that use industrial applications for evaluations. Realistic evaluations require studies conducted in industrial settings, using practitioners as subjects and utilizing industrial applications [49]. On the other hand, studies conducted in a laboratory setting, for e.g. in a research laboratory by a researcher often exercise a high level of control and the setting itself is artificial and therefore lack. Such studies are also important and have their own advantages. For example, results of evaluations in research laboratories can identify what should later be evaluated in an industrial setting.

Table 34 Studies utilizing industrial applications

ID	Research methods	Research context	Subjects	# of studies
A	Case studies, experiment, lessons learned	Industry	Practitioners	15
B	Surveys	Industry	Practitioners	4
C	Case studies	Industry	Researchers	4
D	Case studies	Academia	Researchers	2

The studies presented in the review are classified as dynamic evaluations in industry, static evaluations in industry and evaluations in academia (adapted from [49]). For our purpose, only studies using industrial applications are considered. Dynamic evaluations are either case studies, experiments or lessons learned in industrial context. Studies that are observations, for example surveys are classified as static evaluations in industry. As shown in Table 34, there are two (ID D, Table 33) evaluations in academia, eight static evaluations in industry (ID B + ID C, Table 33) and 15 dynamic evaluations in industry (ID A, Table 33). Of the 46 studies, 33% studies are dynamic evaluations in industry. This number is encouraging in terms of evaluations of technologies related to QR.

Next, to evaluate the strength of evidence, studies that offer high degree of realism are analyzed. By combining the properties 2 - 4 (Context described, Study design and Validity discussed) we can identify the extent to which contexts of the evaluations, design of the evaluations and their validity have been described in the reporting studies. These three properties are combined (See Table 35 below) and the research rigour found in the evaluations reported in the studies is quantitatively assessed as Quality scores. Table 35 shows the scores for research rigour for the papers reporting dynamic evaluations in industry.

Table 35 Research rigour for papers reporting dynamic evaluations in industry

Context described	Study design	Validity discussed	Quality scores	Studies	# of studies
Strong	Strong	Strong	6	[42]	1
Strong	Strong	Medium	5	[2], [13], [18]	3
Strong	Strong	Weak	4	[3], [25], [32], [43]	4
Strong	Medium	Weak	3	[1], [29], [34], [4]	4
Medium	Medium	Weak	2	[17], [35]	2
Weak	Medium	Weak	1	[45]	1

As the table above shows, only one study out of the 15 dynamic evaluations in industry have high scientific rigour. In addition, three studies obtained a score of 5. This means that only four out of 46 studies (9%) contain strong evidence for the effect of the use of technologies in relation to QR and, at the same time offer a high degree of realism. **Table 36** shows the scores for research rigour found in studies reporting static evaluations performed in industry and evaluations in academia utilizing industrial applications.

Table 36 Static evaluations in industry and evaluations in academia utilizing industrial applications

Research context	Context described	Study design	Validity discussed	Quality scores	Studies	# of studies
Industry	Strong	Strong	Strong	6	[23], [41], [44]	3
Industry	Strong	Strong	Medium	5	[46]	1
Industry	Strong	Strong	Weak	4	[9]	1
Industry	Medium	Medium	Medium	3	[8]	1
Industry	Medium	Medium	Weak	2	[28]	1
Industry	Medium	Weak	Weak	1	[26]	1
Academia	Strong	Strong	Strong	6	[14]	1
Academia	Medium	Medium	Medium	3	[33]	1

As seen from the above table (Table 36), four studies obtained a quality score of 6 and one study a score of 5. Thus, half the studies (five studies out of ten) have strong evidence – see Table 36. From Tables 35 and 36, the review identified only nine studies offer strong evidence on the use of technologies in relation to QR.

Analysis: The number of dynamic evaluations in industry with a high degree of scientific rigour is small (four studies). The investigation, however, identified five more studies with a high scientific rigour. These studies used industrial applications and thus provide important material on technologies involving QR. Table 37 shows the findings from the studies.

Table 37 Findings of studies with a high degree of scientific rigour

Study	Process area addressed	Quality score	Summary/Findings
[42]	Elicitation, prioritization & specification	6	The paper presents a lightweight method that gathers relevant stakeholders to elicit, prioritize and elaborate the quality of a software product using quality indicators. The method is adapted from QUPER model [25, 35]. The method has been implemented in four companies. The method uses brainstorming and yellow stickers for elicitation. Yellow stickers pasted on walls are used to add new ideas for incorporation into QR. Quality goals are written on the yellow stickers with a brief description. All quality goals that are written are discussed in a workshop session. In the workshop session, the ISO 9126 quality model is used as a checklist to further elaborate and prioritize quality goals (or QR). QR are prioritized through voting. Voting is done from the viewpoints of importance of QR for the product. The collection of viewpoints thus generates a holistic view. There are several reasons for this. First, the elaborated QR that receives the maximum votes is the most important quality aspect of the product and receives highest priority. Second, viewpoints of all the participants are considered. Third, the participants are asked to consider their viewpoints so that all the elicited QRs are considered during prioritization and voting is done publicly. The findings of the method are: 1) It is better to set quality goals for a product first and then elaborate for project processes. 2) The use of ISO 9126 as a checklist for the identified quality goals is not necessary. 3) Identification of quality indicators is useful. 4) Prioritization and conceiving measures for quality indicators are challenging tasks. The method was developed to fill the need for a lightweight and practical method. The authors point out that other methods [9, 13, 45] for elicitation and

			specification are complex in nature and are not practical. All in all the method offering elicitation and prioritizing practices was found to be promising. The method can improve in developing competence and knowledge building. For example, workshop sessions and brainstorming can foster product related QR development and experience-based knowledge can be utilized in the subsequent projects. The name of the product used for evaluation was kept anonymous. However, the method was applied in four companies with each having at least a dozens of customers. The method can be tailored to software products in different domains.
[2]	Prioritization & metrics	5	The paper presents a survey where key quality characteristics for intranet applications (reliability, functionality and efficiency) are identified. The authors developed three quality metrics: Availability, Failure rate and Normalized failure rate. The validity and usefulness of the metrics were evaluated in an experiment by applying them to five in-house intranet applications. The metrics were found to be useful, practical and economical for measuring the quality of intranet applications. It is economical in the sense that a small effort is needed for deriving measures and the cost of obtaining the metrics is low. Specifically, the metrics provided means to measure the quality of the applications and can be used as a basis for improvement in the development of intranet applications.
[13]	Elicitation & prioritization	5	The paper presents an experience based method to elicit, document, and analyze QR. The method uses workshops to capture QR and questionnaires to prioritize. The method is implemented in an industrial setting. ISO 9126 and IEEE quality standard 830-1998 were used as quality reference models. It is found that collaborative workshops and the use of quality models could be used to define the requirements granularity. The method requires the specification of functional requirements in the form of use cases. The findings of the study are that the experts needed to spend more time on developing use cases for requirements and analyzing but there was a positive return on the invested time. The benefit of the method is improved communication and definition of common ground as a basis for deriving QR. All in all the method was found to be promising for use during the elicitation process.
[18]	Testing	5	The paper presents experiences from usability testing using “wizard of Oz” testing based on a paper prototype in a project using XP. Usability story boards were developed as a means for collaboration with the end user.

			Usability testing turned out to be of far greater value than anticipated. The following benefits were observed: 1) Usability testing gave a tangible vision of what the project needed to achieve. 2) End users were used as subjects for the usability test and this helped the development team to get an idea of what exactly is delivered as prototypes. In addition, the scenarios give frequent interaction with the end users. 3) Increased end user acceptance. 4) The project was delivered on time. To track the business progress the project used burn-down charts. The paper concluded that usability testing using "Wizard of Oz" testing based on paper prototype and user stories did not require formal training. Only common sense and domain experience was needed.
[14]	Elicitation & prioritization	6	The paper presents a method for elicitation and prioritization of security requirements (SQUARE). The method was examined in two case studies and was found to be useful for understanding security requirements. The SQUARE model consists of nine steps. Performing risk assessment of the elicited requirements is an integral part of the method. The method has been implemented in an organization. The details of the organization are kept anonymous. The method shows promising results in helping the company in addressing security requirements. Based on an industry implementation, changes were proposed to the method. SQUARE method and implementation have been described in [90]. Security requirements are documented using a custom made template. Details of the security goals are captured using "misuse cases". Prioritization is done by assigning priority levels to the elicited misuse cases by individual team members and comparing the average of the team's priority levels and those of the client.
[23]	Metrics, specification & testing	6	The paper presents an assessment of aspect-oriented techniques for testing QR. The methodology involves identifying system characteristics to be tested with aspects followed by a description on how to derive test objectives. Cross-cutting system characteristics (QR) and constraints on the system are specified using natural language. From the test objectives, we can formulate test aspects. Test aspects describe the system concerns that are covered - e.g. supervising memory consumption to track performance issues. This resulted in an increase in the overall test coverage. A lack of tool support for aspect-oriented extensions and final code instrumentation in order to weave aspects into the system was observed. However, managing test aspects is an easy step compared to traditional testing. The ability to identify cross-cutting

			testability issues was found to be a merit for aspect-oriented techniques. As the aspects are derived from requirements (i.e. QR), treating aspects as architectural elements harnessed designing verification, thus ensuring a good software architecture.
[41]	Elicitation & prioritization	6	An interview based study to identify how QR are handled in practice
[44]	Prioritization	6	The paper presents a survey where QR from the perspective of software architects are prioritized.
[46]	Elicitation & specification	5	The paper presents a security requirements elicitation approach – SeqReq - that integrates elicitation, traceability and analysis activities. The approach combines the security standard ISO 14508 Common Criteria (CC) [90], the heuristic requirement editor, HeRA tool [91] and UML security extension, UMLSec. Security concerns are expressed as UML stereotypes. Combining the three techniques facilitates tracing and mapping of security requirements to design. The approach is built on five principles (also described as steps): *specific*, *measurable*, *achievable*, *realizable* and *traceable*. The approach has been applied and evaluated in ETSI using the Internet protocol television (IPTV) application. The approach works best when an expert in the field of security requirements engineering participates in the process. This was attributed to the difficulty in identifying security goals and objectives. HeRa could, however, not give full coverage for the security goals. There were also some difficulties when identifying goals from security functional component part of the CC standard. Expertise and experience was needed here. However, the reported advantage of the approach is its tool support (UMLSeq) that is used for tracing from requirements to design and vice versa.

- Only four of the 46 studies reviewed are found to offer strong support that could be used as input for a decision on technology transfer. Another five studies are found to be promising in terms of research and evaluations based on industrial applications. The overall amount of evidence found in the review is low. Only nine studies (two of which are investigations offering insights into terms of research directions) are found to offer strong support for practitioners. By assigning quality scores to the individual studies, we found that many of the studies are weak in terms of validity. Limitations of the study were not discussed and the credibility of the studies is weak. The study design of the majority of the studies was not discussed in detail, indicating a lack of a proper set of guidelines when implementing the technologies described. That is, the articles do not describe the procedures and how the evaluations were conducted. Looking at the evaluations one does not get a clear understanding of the factors that influence the implementation of a technology in a real setting. The cause-effect of the factors producing a certain result is not known.

- The nine studies described in Table 37 offer strong decision support material. The process areas are well covered and there is a strong focus on solutions offered for elicitation, prioritization and specification (see Table 37). Natural language is preferred when specifying QR ([14, 42]). The use of UML in specifying QR is also found to be promising [46] but there is a need for tool support.

- Testing was found to be a weak spot in handling QR. This result is in line with the observation in [41]. The review identified only one study that offers strong evidence that could be used for technology transfer, i.e. technologies that can be used for testing QR. The paper [18] presents experiences from testing usability in an agile environment. Other quality characteristics that received attention and for which there is strong decision support material are security [2, 13, 14, 46] and performance [2, 23]. Usability and performance requirements are found to be the most important QR [41, 44].

- Among the process areas, elicitation and prioritization have studies that provide good decision support material. Other process areas lack strong evidence but nevertheless, having sound elicitation and prioritization practices and support is a step in the direction of improving the state of the art and industry practices. Elicitation and prioritization constitute the first phases in RE. Here there is good decision support for elicitation and prioritization but more research effort should be invested in understanding real-life industry practices in specifying and testing QR and how QR are measured. It is found that the state of the art suffers from insufficient scientific rigour. Thus, it is difficult for practitioners to use the material when they need decision support for the adaptation of technologies that can be used to handle QR.

- The use of quality standards like ISO 9126 and IEE standards is found to be a common practice. Since there are several viewpoints on QR definitions, using standards give support to practitioners in identifying initial quality goals. Brainstorming and voting mechanism using viewpoints were found to be a common practice in identifying quality goals.

CHAPTER 7
Industrial interviews results

7.1 Participant C at company gamma

As mentioned earlier (Chapter 5.6), three persons agreed to participate in the interviews. The questionnaire was sent to the participants via e-mail. However, participant C replied that the company develops software products, but the requirements are non-negotiable, coming from the customers in the form of an RFQ: Request for proposal. Moreover, the practice of identifying and incorporating QR was not handled in a systematic manner and therefore the interview questions were not answerable from the perspective of the company's RE. In the delivered system, the requirements are just marked as compliant or non-complaint to the RFQ. Thus, the project team does not have the possibility to influence the requirements. The company therefore has no established process in relation to QR.

The data from participant C draws attention to a need for further investigation of companies where technologies or mechanism that supports incorporating QR are not used. This apparently is not due to a lack of technologies but to the way requirements are decided by the customer. Thus, the project team is not allowed to negotiate the requirements given by customers. For example, customers say they want a system in a certain way and the project team has to deliver it. The project team is not allowed to influence or modify the requirements.

7.2 Participant A at company alpha

Participant A is a requirements engineer working for an organization with more than 14000 employees. The company develops phone devices, media services, application servers etc. The product management is responsible for handling requirements in the company. Persons that handle requirements at the company include stakeholders, requirements engineers, system analysts, product and project managers and architects. The effort involved in handling requirements is large and a special process/approach is used for this. The following steps are used in the RE process:

1. New requirements are provided by stakeholders.
2. The RE team accepts or rejects new requirements by analyzing the validity and value of the requirements together with product and service management, using brainstorming sessions.
3. Accepted requirements are sent to the software architects who suggest high level design solution.

4. The solution is reviewed by product management and accepted or rejected based on the architects' suggestions.
5. Rejected solutions are sent back to RE department which keeps on revising the solutions until one is accepted.
6. Accepted solutions are sent to the development team for effort estimation.
7. A delivery version for the requirements is planned by the product management.

The RE department of the company considers it an important part of the company when it comes to including QR into projects as they define customer satisfaction for new products and releases. The company has identified and incorporated more than 20 types of QRs (e.g. security and traceability, See Appendix B) in the past.

QR are prioritized based on the customer business case and aims at profit maximization and cost minimization associated with QR. However, it is a common practice to ignore or forget the elicited QR during the development. They are not considered for further development and are thus not present in the final product. The reason for this is time limitations, overload from errors and work related to handling customer complaints. Changes in requirements are implemented and delivered to customers in subsequent releases.

Specifying QR is found to be challenging to the company since the requirements attached to quality characteristics from the customers are abstract. They come in the form of complaints, often as a few words or sentences. The complaints are analysed by the RE department and product management in collaboration with the customers. Interviews are used for this purpose. RE department and product management interview customers to get more details so that they can understand the customers' problems and business needs.

The company has no particular way of specifying QR, neither a technique nor a technology is used to address the quality specification problems. The problem of specification is related to the type of software products that are developed. The company has a wide variety of products in their portfolio and a wide variety of customers. The company lacks a technique that can be tailored to specify QR for their projects – partly due to the aforementioned product variation. Therefore, QRs are described using natural language even though the need for a better way to describe QR has been identified by the company. The challenge of QR specification is customer-centric. That is, customers are supposed to specify the QR. It is, however, found that customers do not always communicate their business needs in understandable way. The RE department and Product management face the problem of understanding customers' business needs which requires good customer contact and domain knowledge when details are needed.

Surprisingly enough, the participant reports that there are no measures collected for QR. Quantification, as reported by the participant, is an estimate of the cost of success/failure associated with the implementation of a particular QR. Thus, measurability is associated with identifying the value impact of a QR. There seems a clear indication of a practice for identifying which of the elicited QR generate value to the customer.

There are several challenges when testing QR. Firstly, a lack of testing technologies was identified and secondly, providing performance index to the prioritized QR. Performance indices are used in the project practices as they give guidelines to testers so that they can identify the quality level of QR in the product. Performance indices are used to see if the required quality is present in the product. As reported by the interviewee, there is a need for techniques that can be used to test QR. Testing is at the present done manually using freeware tools.

It is surprising to find that in this large company there is a clear need for technology transfer. The interview report of the company participant is encouraging in the sense that a need for new technology evaluation and thereby adoption is identified. At the same time, however, it is little discouraging that a large company like Alpha with a large product portfolio is lacking technical and technology expertise for handling QR.

7.3 Participant B at company beta

Participant B is a systems development manager in a small company - less than 50 employees. The company mainly develops web applications for customers. The participant has 30 years of experience in handling requirements. There are three system developers who are involved in RE in the company. The RE process is a variant of the Rational Unified Process (RUP) where the requirements specification is developed in close cooperation with the customers. QR such as accessibility, integrity, presentability, usability, future business potential and timeliness tend to be incorporated in almost every project. Some of the elicited QRs are not included in the projects, meaning they are dismissed and not implemented. The reasons for dismissal of QR, as reported by the participant, are cost and time factor. The consequences of incorporating QR tend to be very high in terms of cost and are difficult to afford. Some QRs are dismissed/removed because of lack of knowledge and information on how to test them – for e.g. lack of knowledge on how to identify the right amount of quality that should be present in the product.

The customers need be active in the process of selecting candidate QR for incorporation and prioritization. The company has, however, no particular or specialized technique for the final selection of QR for implementation. The QR specification is, however, developed in corporation with customers.

The QRs are specified using natural language. A value driven approach is used for quantification/measurability of QR. That is, a QR is measured by associating it with its value to the customer. QR measurement is done by collecting measures which form the basis for testing. For example, measures like MTBF and processing time are collected to measure Efficiency. Validation of QR is performed by system developers and the results are discussed with the customers. However, a large and varied quality related nomenclature (e.g. performance, efficiency) is identified as a problem. Specific quality characteristics related terms, for example response time and processing time are used instead of the more general term efficiency.

CHAPTER 8

Validity threats

Four types of validity threats are commonly discussed in literature [82, 83,91]. These are *Conclusion validity*, *Internal validity*, *Construction validity* and *External validity*.

8.1 Conclusion validity

Conclusion validity helps to know the factors that can affect the reliability of results and conclusion. In relation to SLR, conclusion validity depends on SLR design and execution. In SLR design, publication bias is a major threat to study validity. This threat refers to the possibility to generalise the results. Four scientific databases are selected as search venues: IEEE Explore, ACM Digital Library, SpringerLink and ScienceDirect. These four databases are premier venues for literature search and paper retrieval. Papers with research significance published in several high quality journals and conference papers are available in these databases. In addition, Inspec/Compendex, a reference indexing venue was used to search papers from SpringerLink and ScienceDirect.

In the second phase conducting the systematic search, a manual search was used on both of these databases so as to see if papers with relevance to the research questions of the thesis had been left out. Only four articles papers were included in this way. The source of papers was not limited to a particular set of publishers, authors or conference proceedings. Thus, the publication bias was reduced. To reduce the occurrence of errors, or retrieval of irrelevant papers caused by the formulation of search strings, categories of keywords were formed separately. Search keywords are derived from observation of terminologies identified in a large variety of papers on RE.

Search keywords related to the individual quality characteristics such as usability and performance were not included. This was done after a pilot search in which quality characteristics terms like usability and performance were included. This resulted in too many irrelevant results and therefore quality characteristics terms were dropped from the final search string used in the review. This was a necessary trade-off as the trial search gave a high number of initial results for review. Most of the results of the trial search were found to be irrelevant to the goal of the review. This would have resulted in large amount of extra work and have been a difficult task. It is also found that several articles that focused on quality characteristics like performance, efficiency have been obtained from the refined search string thus raising confidence on the obtained set of primary studies.

The search string was applied on *Title* and *Abstract.* This is because if a papers central theme is about a particular technology for QR, it ought to have description in the paper’s abstract. Threats occurring due to the formulation of search string and searching in the databases were thus kept at a minimum.

Inclusion and exclusion criteria are formulated to select papers relevant to research questions and to the overall goal of the thesis. The inclusion and exclusion process was tested at a later stage by comparing the results with the papers presented in [49]. As [49] presented articles in RE from a technology transfer perspective, this helped us to identify missing papers. Only one paper [45] was found in [49]. Data extraction was challenging and if not properly controlled could cause several threats. This challenge was attributed to the way technology evaluations were presented. Studies describing technologies and evaluations should be understandable and contain detailed descriptions of their applicability. While considering a technology for adoption, we need studies that contain a description of a setup consisting of a sequence of steps to be followed, thus making a replication of the study possible.

There can be similar threats to the conclusion validity for industrial interviews stemming from the design and execution of the interview questionnaires. In order to cover the objectives of the thesis questionnaire, we designed and performed a preliminary testing to validate its coverage. This was done by consulting the supervisor and two master thesis students in Sweden who had experience in performing industrial interviews.

The small amount of information collected at the execution level of the interviews can also pose a threat to conclusion validity. The initial plan of the interviews was to conduct them as telephone interviews. This was, however, changed to an e-mail based data collection due to requests from the participants. A face-to-face interview or telephone interview results would most likely have resulted in eliciting more information.

Based on the answers to the first set of questions, a new batch of questions were added and sent to the participants so to elicit more information. In this way threat the threat to validity due to the way interviews were executed is reduced. Sending new batches of questions allowed to elicit information which otherwise could not have been collected.

8.2 Internal validity

Internal validity refers to how we establish a casual relationship between treatment and outcome when drawing conclusions from the study. The studies reviewed in the thesis differed in quality. In a topic like technology adoption, it is essential to find the potential of a particular technology for adoption in terms of how they scale to realistic settings. Therefore, to identify sound technologies for adoption, a rigorous quality assessment was performed on the studies. In order to find promising technologies presented in studies, studies offering strong evidence and scientific rigour are identified. In order to identify the strength of evidence, the method adopted should be a valid one. In this thesis we adopted the method followed by Ivarsson [49]. We found the method applicable to our thesis because of the commonalities in Ivarsson's work [49] and ours. These commonalities are identifying RE technologies, which in our case are focussed on technologies for QR, and finding decision support material based on how the evaluations were presented.

A possible threat to the internal validity is the selection of companies and participants for the interviews. Initially this was a challenging task as the author did not have any industrial contacts. However, two software development companies were identified through the supervisor. The third company, identified by the author, is a large and well established company. The participants who agreed for the interviews have a long experience in software development, thus reducing the threats to the validity of the study.

8.3 Construct validity

Construct validity refers to the relation between theory and its application. A possible threat to construct validity was the exclusion of studies from the review. As the author was the only reviewer of the studies, it was not possible to control the validity of data extraction using discussions to achieve a consensus or using a statistical test – e.g. kappa tests [92]. The results were, however, always communicated to the supervisor for suggestions and comments. In order to improve the validity of data analysis and reduce researcher bias a well defined review protocol is needed. For this reason we developed the data extraction based on earlier SLRs. A strategy that is commonly used in SLR is to have the extracted data cross-check by several researchers but this could not be done in this thesis. To counter this problem and check the consistency of the data extraction, we performed two rounds of data extraction. We did not find differences in the extracted data. Thus, we achieved full control over the data presented in the articles. We found that the extracted data were consistent, thereby increasing the validity of the data analysis. Moreover, only empirical studies or studies with empirical evaluation were included. Thus, research papers presenting new technologies and insights without empirical evidence were left out. This is not per se a threat to the review, but with more empirical evaluations there will be eventually more technologies to review and find useful results for practitioners.

Interviewing a single person will not give complete information of the QR status in the company and interviewing several persons from the same company will give a better picture of this particular company. The information will, however only describe the reality for a this company and therefore not give a broader picture of reality in the industry. In order to avoid this threat, participants from several companies were selected for participation in the industrial interviews.

8.4 External validity

External validity refers to the ability to generalise the findings beyond the actual study. One threat to external validity is the analysis of the results from the industrial interviews. As the data is collected from only three companies, there are problems with generalizing the results to the whole software industry. The choice of participants for industrial interviews was limited because of the scope of this research. Therefore, collecting data from more companies as part of future work will improve the possibilities for generalizing the results.

8.5 Overall credibility of the thesis

To the best of our knowledge our study is the first of its kind in reviewing and finding decision support material for QR technologies. The SLR commenced in May, 2010 and by then similar studies, or a review on RE research for QR was not published. The threats to validity concerning SLR can be classified into the following three: Publication bias, Identification of primary studies and Data extraction. Threat to validity due to Publication bias is small. We did not restrict our search to a particular set of journals or conferences, but looked in premier search venues that have high quality scientific, peer-reviewed papers.

In order to decrease the threat stemming from the identification of primary studies, we developed our search string by including a large variety of keywords. For example, several synonyms for the term “Quality Requirements” were used. After a trial search we excluded the search terms for individual QR or quality characteristics like performance, efficiency.

To identify and understand industrial practices we performed industrial interviews. To improve the validity of the industrial data, companies from three geographical locations were interviewed. This allowed us to sample on a wider scale, thus improving the possibility of generalization. A possible threat is that we could not perform telephone interviews. Instead, we received answers to the questionnaire via e-mail. We believe telephone interviews could fetch much more information relevant to our goal. However, this was not possible due to the accessibility of the interviewees. Nevertheless, we believe that the elicited data from the companies is important in understanding how QR are handled, thus achieving the objectives of this review.

All in all we believe we handled the threats well, thus improving the validity of the study and increasing the amount of confidence we can place on the obtained results, both in the SLR and in the industrial interviews.

CHAPTER 9
Discussion

This chapter presents our discussion combining the results of SLR and industrial interviews, our conclusions and directions for future work.

9.1 SLR and industrial interviews

In our SLR we identified 46 studies that reported empirical evaluations of the presented technologies. Of these 46 studies seven studies are of an investigative nature. The themes addressed by the studies include, among other things, identification of most important QR, views of architects' on QR, identification of metrics for intranet applications and objectives concerning elaboration of security requirements (See Table 7). We mapped the rest of the evaluations to the process areas: Elicitation, Prioritization, Specification, Metrics and Testing. Of the reviewed 46 studies (39 empirical evaluations and 7 empirical investigations), we found only nine evaluations with a high strength of evidence (two of which are empirical investigations). These findings are described in Table 37, Chapter 6.5.

Looking at evaluations reported in these nine studies, five studies address the process area Elicitation ([13, 14, 41, 42, 46]), six studies address the process area Prioritization ([2, 13, 14, 41, 42, 44]), three studies address Specification of QR ([23, 42, 46]) while two studies addressing Metrics ([2, 23]) and two addressed Testing ([18, 23]). Looking at the number of studies addressing each process area, there is a higher number of studies with high strength of evidence on Elicitation and Prioritization while there are just a few studies on Metrics and Testing (two studies each).

Looking at the responses from the participants in the industrial interviews, we found no use of existing technologies (methods, techniques, models or frameworks) to handle QR. Possible reasons for this can be lack of management support, lack of possibilities for negotiations in requirements elicitation, lack of knowledge and organizational factors that include budget and time constraints.

The overall observations of our SLR and industrial interviews are summarized below.

- In our SLR we identified a plethora of technologies that have been empirically evaluated in academic and industry setting. A common observation in many of the evaluations is a lack of realism. Many evaluations were not performed on industrial examples. That is, many of the evaluations were not performed in realistic settings, i.e. using industrial applications and practitioners as the subjects of the evaluations. Lack of realism can hamper the adoption of the technology. This is because when considering a technology for industrial practice, factors concerning the actual use and usability, i.e. possible benefits and limitations and applicability must be evaluated. Thus, we need to perform evaluations in realistic settings. We identified only six studies that scale to realistic settings and another eight studies that were performed by researchers and using industrial applications. Although fourteen evaluations used industrial applications, only nine studies demonstrated high scientific rigour. Scientific rigour is the extent to which an evaluation is described, i.e. context of the evaluations, how the evaluation is designed and conducted, and describing how valid the evaluation is, possible benefits and limitations. These factors demonstrate how valid a technology is for technology adoption.

- Customer influence/power is an important factor in deciding how RE is practised – e.g. QR - and for deciding technology adoption. For example, we observed customer influence on RE practices at company Gamma. The project team at company Gamma does not have the possibility of influencing the requirements. Generally customers give an overview of what they want in the product. It works as a contract – we want these particular set of requirements in the product, otherwise we go elsewhere. Failure to deliver customers' requirements results in business loss. This way the requirements in most cases cannot be negotiated and the product management team cannot does not have the power to influence or change customer' requirements.
- We also found challenges in handling QR at company Alpha. These challenges are due to lack of knowledge in testing QR, identifying measures for QR, specifying and elaborating QR in an understandable way for development.
- Our SLR identified potential technologies that can be used to addresses the identified challenges: Aspect-oriented techniques [23] to specify and test QR, Security requirements elicitation approach, SecReq [46] to integrate elicitation, traceability and analysis activities of security requirements. Security is one of the most used and important QR considered for implementation in the case of Company Alpha and our SLR identifies SecReq as a potential candidate technology to consider for adoption for this company. SQUARE methodology [14] is also found to be promising in handling security requirements. In addition, our SLR found it to be a promising methodology for handling the elicitation and prioritization of security requirements and to perform risk assessments of the elicited requirements. Companies that are in need of a potential technology to handle security requirements should consider SecReq and SQUARE.
- Budget and time constraints are common to all the companies in handling QR. We found that some of the candidate QR are not implemented due to time and budget constraints. This can be attributed to a lack of systematic way of handling QR. For instance, in the case of company Beta we observed there is no systematic set of practices that are used for handling QR in projects, which makes it difficult to implement them. A possible solution to this challenge is to have sound elicitation and prioritization practices. This calls for a lightweight, practical method. Our SLR identified a lightweight method [42] that helps practitioners to elicit, elaborate and prioritize QR. This method has active roles for all relevant stakeholders since they have to participate in brainstorming sessions where QR relevant for the project are elicited and prioritized. We find the lightweight method [42] suitable to the companies we interviewed since it does not require the use of a new set of tools or change of the existing organizational structure and does not require huge budget and

effort to implement. This method benefits the elaboration of the elicited QR, therefore enabling practitioners to analyze the amount of effort and technical expertise that will be necessary to implement the elaborated QR.

- Another possible way to mitigate the challenge of cost constraints is to reuse the knowledge of implemented QR in past projects. We also identified a potential method for this in our SLR - an experience based method [13] that facilitates using past results in elaborating QR goals, elicitation and prioritization of QR for the current project. This method requires that the use cases for the functional requirements are available. Appropriate QRs are selected based on the functional requirements, and then elicited and elaborated based on the use cases for the functional requirements. The experience method [13] can be complemented by the lightweight method [42] and is recommended as a candidate technology that offers a systematic way of handling QR in software projects.
- Through our SLR and industrial interviews we identified a considerable gap between the state of research and industrial practices. That is, the models we find in our SLR are not found in industrial practices. Software development needs methods that are practical and scalable to industrial needs and the research efforts should be focusing on inventing technologies to solve industry needs. Most importantly, technologies should be evaluated in realistic settings. Technologies evaluated in realistic settings will show the strength of evidence and possible benefits and limitations of the technologies to the practitioners.
- We observed that many papers in our SLR suffer from scientific rigour. It is important to describe the design of the evaluation and the validity of the studies. Empirical studies can benefit from the guidelines provided by Kitchenham et al. [93] and the checklist used by Dybå and Dingsøyr [87]. All in all, empirical studies reporting evaluations should contain: (1) *description of the context* of the evaluation that helps the reader easier to understand where the evaluation was conducted – e.g. industry or research laboratory, conducted in corporation with industry as part of a technology transfer effort, product used for the evaluation, customer segment the product is targeted to. (2) *description of the design* of the evaluation and research method(s) used, giving a description of the suitability of research method(s) used, description of data collection and analysis procedure(s). (3) *description of the validity claims* for the evaluation, describing the validity of the evaluation, possible factors that could have affected the results of the evaluation, lessons learned and possible directions in future work.
- The encouragement for a technology transfer effort should come from industry also. In order to mitigate industry problems and challenges, it is important to communicate with researchers to develop practical and scalable solutions. Industry practices in handling QR and research efforts should go hand in hand with the efforts of academia and enable both parties to reap benefits.

9.2 Conclusions

This thesis presented a systematic literature review of evaluated QR technologies plus three industrial interviews. The aim is to provide decision support material that can be used when a company wants to (1) incorporate quality requirements technologies for elicitation, prioritization, specification, measurement and testing of quality requirements, (2) to identify existing support for technology transfer and adoption, i.e. to examine the evaluations of the presented technologies and (3) see to what extent practitioners can benefit from and use the reporting of the presented evaluations to handle quality requirements. To identify industrial practices we conducted interviews with three companies. The following are the main findings of this thesis:

- **Low evidence for technology transfer support:** Only nine studies (seven evaluations and two empirical investigations) out of 46 offer high strength evidence. The majority of the studies reporting evaluations were performed rather poorly. Evaluations suffer from incomplete description of study design and validity (See Table 13). We also observed low evidence for information needed to support decisions related to technology transfer.
- **Gap between research and practice:** First, we observed a large gap between research and practice. By reviewing the technology evaluations we observed that many of the presented technologies were not tried out in industry, i.e. they were not performed in realistic settings using industrial applications. From the data obtained from our industry participants, we found that none of the technologies (models, techniques, methods, or approaches) found in our SLR are used in industry. Second, we observed challenges in relation to handling quality requirements. These include lack of knowledge for how to test quality requirements – e.g. lack of knowledge in testing quality requirements and lack of systematic ways of specifying quality requirements. Quality requirements are often not clearly stated, i.e. in a way understandable to the RE teams. This makes elaboration of requirements a challenging task. We observed a need for technologies that facilitate a systematic way of handling quality requirements – i.e. eliciting, prioritizing and elaborating quality requirements. We found that many of the quality requirements that are elicited are later rejected by product management, i.e. not implemented because of time and budget constraints. Incorporating quality requirements late in the projects have consequences in terms of increases in cost and time. Therefore, a systematic way of handling quality requirements is recommended.
- **Need for further research and evaluations:** By examining the evaluations presented and the overall evidence available to support technology transfer, we find a strong need for further research and evaluations. The ultimate goal of research should be to understand industry problems and present technologies that are practical. The presented technologies should be tried in industrial settings. Software engineering is a young and dynamic field. To facilitate the transfer of research results from academia to industry, management support is necessary.

9.3 Future work

After performing a systematic literature review and industrial interviews, we identified a set of interesting directions as part of future work.

- **Performing evaluations in realistic settings**: To provide results that support technology adoption, more evaluations should be performed using industrial applications.
- **Need for presenting more technologies:** In our opinion, there is too little technology evaluations in the process area Testing. This is in line with the observations in our industrial interviews. Only two studies offer strong evidence. Interestingly, we found one study, reporting "Wizard of Oz" testing in agile environment, which offers support for technology transfer. It is, however, not clear from our review whether this testing practices and technologies are adopted for agile based software development. This is an interesting area to explore further.
- **Factors involved in handling Quality requirements:** Our thesis has been aimed at the process areas Elicitation, Prioritization, Metrics, Specification and Testing. In our industrial interviews we identified challenges that include cost and time constraints. A possible direction for further work is to explore factors affecting cost estimation and cost for development related to quality requirements and how they are intertwined with functional requirements.
- **Empirical studies to investigate practices to handle Quality requirements:** As part of our thesis we investigated only three companies, of which one company did not need practices for handling quality requirements and therefore could not participate further in our interviews. To this end, our results encourage further empirical studies to investigate industrial practices on a large scale. This could be performed as surveys where data on practices and technologies adopted can be collected and thus used to identify the state of industrial practice in this area. Such surveys can identify the industrial needs, resources required to handle quality requirements and finally produce practical and economical solutions.
- **Lack of tool support:** We found little evidence on the evaluations of tools for handling quality requirements. Only three studies reported evaluations of tools - none of them provided strong evidence and the tools thus cannot be recommended for adoption. There is a need for more evaluations of tools and research on tool support for handling quality requirements.
- **Borrowing concepts from general requirements engineering:** Requirements engineering is a broad area. We did not investigate the suitability of technologies used for functional requirement, nor has such a study been performed elsewhere. For example, prioritization techniques like AHP, techniques like RAM [117] have been proposed and evaluated for functional requirements. The suitability of such techniques should also be investigated for quality requirements. The research can start by using experiments in academia involving researchers and students as subjects before starting to do experiments in an industrial setting. On the other hand, these experiments are subject to the willingness of the companies as experiments

tend to be costly. More case studies can be performed using industrial examples as case studies offer a high degree of realism.

CHAPTER 10
References

SLR references

[1] M. Ruthford and J. Ramey, "The design response to usability test findings: a case study based on artifacts and interviews," *18th Annual Conference on Computer Documentation. ipcc sigdoc 2000. Technology and Teamwork. Proceedings. IEEE Professional Communication Society International Professional Communication Conference and ACM Special Interest Group on Documentation Conferenc*, 2000, pp. 315-323.

[2] H. Leung, "Quality metrics for intranet applications," *Information & Management*, vol. 38, 2001, pp. 137-152.

[3] L.M. Cysneiros, J.C. do Prado Leite, and J. de Melo Sabat Neto, "A Framework for Integrating Non-Functional Requirements into Conceptual Models," *Requirements Engineering*, vol. 6, 2001, pp. 97-115.

[4] K. Kang, "Quality requirement elicitation for the architecture evaluation of process computer systems," *Proceedings Eighth Asia-Pacific Software Engineering Conference*, 2001, pp. 335-340.

[5] M. Hassenzahl, R. Wessler, and KC, "Exploring and understanding product qualities that users desire," *sans frontières. Actes*, 2001, pp. 95-6.

[6] B. Boehm, T. Rodger, and M. Deutsch, "Applying WinWin to quality requirements: a case study," *Proceedings of the 23rd International Conference on Software Engineering. ICSE 2001*, IEEE Comput. Soc, 2001, pp. 555-564.

[7] K. Kang, "Quality requirement elicitation for the architecture evaluation of process computer systems," *Proceedings Eighth Asia-Pacific Software Engineering Conference*, 2001, pp. 335-340.

[8] D. Ameller and X. Franch, "How Do Software Architects Consider Non-Functional Requirements : A Survey," 2010, pp. 276-277.

[9] L. Cysneiros and J. do Prado Leite, "Nonfunctional requirements: from elicitation to conceptual models," *IEEE Transactions on Software Engineering*, vol. 30, 2004, pp. 328-350.

[10] A. Gregoriades and A. Sutcliffe, "The system reliability analyser tool," *Requirements Engineering*, vol. 10, 2004, pp. 63-80.

[11] P. Donzelli, "A goal-driven and agent-based requirements engineering framework*," *Requirements Engineering*, vol. 9, 2004, pp. 16-39.

[12] Y. Zou, "Incorporating Quality Requirements in Software Migration Process," *Eleventh Annual International Workshop on Software Technology and Engineering Practice*, 2004, pp. 175-185.

[13] J. Doerr, D. Kerkow, T. Koenig, T. Olsson, and T. Suzuki, "Non-Functional Requirements in Industry - Three Case Studies Adopting an Experience-based NFR Method," *13th IEEE International Conference on Requirements Engineering (RE'05)*, 2005, pp. 373-384.

[14] N.R. Mead and T. Stehney, "Security quality requirements engineering (SQUARE) methodology," *ACM SIGSOFT Software Engineering Notes*, vol. 30, 2005, p. 1.

[15] L. Chung and S. Sapakkul, "Capturing and Reusing Functional and Non-functional Requirements Knowledge: A Goal-Object Pattern Approach," *2006 IEEE International Conference on Information Reuse & Integration*, 2006, pp. 539-544.

[16] P. Donzelli and V. Basili, "A practical framework for eliciting and modeling system dependability requirements: Experience from the NASA high dependability computing project," *Journal of Systems and Software*, vol. 79, 2006, pp. 107-119.

[17] M. Johnson, L. Williams, and E. Maximilien, "On Agile Performance Requirements Specification and Testing," *Agile 2006 (Agile'06)*, 2006, pp. 47-52.

[18] G. Meszaros and J. Aston, "Adding Usability Testing to an Agile Project," *Agile 2006 (Agile'06)*, 2006, pp. 289-294.

[19] F. Aburub, M. Odeh, and I. Beeson, "Modelling non-functional requirements of business processes," *Information and Software Technology*, vol. 49, 2007, pp. 1162-1171.

[20] J. Chen, Y. Lu, X. Xie, and W. Zhang, "Testing approach of component security based on dynamic monitoring," *Second International Multi-Symposiums on Computer and Computational Sciences (IMSCCS 2007)*, 2007, pp. 381-386.

[21] Y. Fei and Z. Xiaodong, "An XML-Based Software Non-Functional Requirements Modeling Method," *2007 8th International Conference on Electronic Measurement and Instruments*, 2007, pp. 2-375-2-380.

[22] T. Hazeem, A. Balushi, P.R. Sampaio, and D. Dabhi, "ElicitO : A Quality Ontology-Guided NFR Elicitation Tool," 2007, pp. 306-319.

[23] J. Metsa, M. Katara, and T. Mikkonen, "Testing Non-Functional Requirements with Aspects: An Industrial Case Study," *Seventh International Conference on Quality Software (QSIC 2007)*, 2007, pp. 5-14.

[24] E. Niemela and a. Immonen, "Capturing quality requirements of product family architecture," *Information and Software Technology*, vol. 49, 2007, pp. 1107-1120.

[25] B. Regnell, M. Höst, and R.B. Svensson, "A Quality Performance Model for Cost-Benefit Analysis of Non-functional Requirements Applied to the mobile handset domain," *Requirements Engineering: Foundation for Software Quality. Proceedings 13th International Working Conference, REFSQ 2007*, 2007, pp. 277-291.

[26] R. Savola and K. Karppinen, "Practical Security Testing of Telecommunications Software--A Case Study," *The Third Advanced International Conference on Telecommunications (AICT'07)*, 2007, pp. 7-7.

[27] H. Siy, P. Aryal, V. Winter, and M. Zand, "Aspectual Support for Specifying Requirements in Software Product Lines," *Early Aspects at ICSE: Workshops in Aspect-Oriented Requirements Engineering and Architecture Design (EARLYASPECTS'07)*, 2007, pp. 2-2.

[28] C.C. van Waveren, "A process framework for customising software quality models," *Africon 2007*, 2007, pp. 1-8.

[29] J. Winter, K. Ronkko, M. Ahlberg, M. Hinely, and M. Hellman, "Developing Quality through Measuring Usability--The UTUM Test Package," *Fifth International Workshop on Software Quality (WoSQ'07: ICSE Workshops 2007)*, 2007, pp. 2-2.

[30] L. Zhu and I. Gorton, "UML Profiles for Design Decisions and Non-Functional Requirements," *Second Workshop on Sharing and Reusing Architectural Knowledge - Architecture, Rationale, and Design Intent (SHARK/ADI'07: ICSE Workshops 2007)*, 2007, pp. 8-8.

[31] L. Duboc, E. Letier, D.S. Rosenblum, and T. Wicks, "A Case Study in Eliciting Scalability Requirements," *2008 16th IEEE International Requirements Engineering Conference*, 2008, pp. 247-252.

[32] C. Ho, L. Williams, and B. Robinson, "Examining the Relationships between Performance Requirements and “Not a Problem” Defect Reports," *2008 16th IEEE International Requirements Engineering Conference*, 2008, pp. 135-144.

[33] H. Kaiya, T. Sato, A. Osada, N. Kitazawa, and K. Kaijiri, "Toward quality requirements analysis based on domain specific quality spectrum," *Proceedings of the 2008 ACM symposium on Applied computing - SAC '08*, 2008, p. 596.

[34] K.V. Prasad, T.J. Giuli, and D. Watson, "The Case for Modeling Security, Privacy , Usability and Reliability (SPUR) in Automotive Software," *Model-Driven Development of Reliable Automotive Services. Second Automotive Software Workshop, ASWSD 2006. Revised Selected Papers*, 2008, pp. 1-14.

[35] B. Regnell, R.B. Svensson, and T. Olsson, "Supporting Roadmapping of Quality Requirements," *IEEE Software*, vol. 25, 2008, pp. 42-7.

[36] M.B. Swarup and P.S. Ramaiah, "An Approach to Modeling Software Safety," *2008 Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, 2008, pp. 800-806.

[37] B. Cho, H. Youn, and E. Lee, "Software Dependability Analysis Methodology," *Work*, 2009, pp. 580-593.

[38] S.L. Ghiron, S. Sposato, C.M. Medaglia, and A. Moroni, "NFC Ticketing: A Prototype and Usability Test of an NFC-Based Virtual Ticketing Application," *2009 First International Workshop on Near Field Communication*, 2009, pp. 45-50.

[39] A. Marback, S. Kondamarri, and D. Xu, "Security test generation using threat trees," *2009 ICSE Workshop on Automation of Software Test*, 2009, pp. 62-69.

[40] Q.L. Nguyen, "Non-functional requirements analysis modeling for software product lines," *2009 ICSE Workshop on Modeling in Software Engineering*, 2009, pp. 56-61.

[41] R.B. Svensson, T. Gorschek, and B. Regnell, "Quality Requirements in Practice: An Interview Study in Requirements Engineering for," *Requirements Engineering: Foundation for Software Quality. Proceedings 15th International Working Conference, REFSQ 2009*, 2009, pp. 218-232.

[42] J. Vanhanen, M.V. Mäntylä, and J. Itkonen, "Lightweight Elicitation and Analysis of Software Product Quality Goals: A Multiple Industrial Case Study," *2009 Third International Workshop on Software Product Management*, 2009, pp. 42-52.

[43] E. Weyuker and F. Vokolos, "Experience with performance testing of software systems: issues, an approach, and case study," *IEEE Transactions on Software Engineering*, vol. 26, 2000, pp. 1147-1156.

[44] E. Johansson, A. Wesslen, L. Bratthall, and M. Host, "The importance of quality requirements in software platform development-a survey," *Proceedings of the 34th Annual Hawaii International Conference on System Sciences*, IEEE Comput. Soc, 2001, p. 10.

[45] A. Herrmann and B. Paech, "MOQARE: misuse-oriented quality requirements engineering," *Requirements Engineering*, vol. 13, 2007, pp. 73-86.

[46] S.H. Houmb, S. Islam, E. Knauss, J. Jürjens, and K. Schneider, "Eliciting security requirements and tracing them to design: an integration of Common Criteria, heuristics, and UMLsec," *Requirements Engineering*, vol. 15, 2009, pp. 63-93.

Other references

[48] M. Scho¨nstro¨m, Software development as knowledge work – a conceptual framework, in: J. Hedman, T. Kalling, D. Khakar, O. Steen (Eds.), Lund on Informatics, CBS Press, Copenhagen, 2005, pp.132–155.

[49] Ivarsson M., and Gorschek T., “Technology Transfer Decision Support in Requirements Engineering Research: A Systematic Review of REj,” Requirements Engineering journal, Vol. 14, Number 3, 2009, pages 155-175.

[50] Wohlin C, Aurum A, Engineering and Managing Software Requirements. Springer, New York, NY, 2005.

[51] Kotonya G, Sommerville I, Requirements engineering – processes and techniques, John Wiley & Sons UK, 1998.

[52] Broy, M., “Requirements engineering as a key to holistic software quality,” Berlin, Germany, Springer-Verlag, 2006.

[53] Withalm, J., “Requirements engineering-most important for improving the software quality,” Dublin, Ireland, Irish Quality Association, 1996.

[54] Sadraei, E., A. Aurum, et al. "A field study of the requirements engineering practice in Australian software industry," Requirements Engineering **12**(3): 145-162, 2007.

[55] Schneidewind, N.F., "Report on the IEEE standard for a software quality metrics methodology (draft) P1061, with discussion of metrics validation," Software Engineering Standards Application Workshop, 1991., Fourth , vol., no., pp.155-157, 20-24 May 1991.

[56] Glinz, M., “On Non-Functional Requirements,” In: 15th IEEE International Requirements Engineering Conference (RE 2007), pp. 21–26, 2007.

[57] Paech, B., Kerkow, D., “Non-Functional Requirements Engineering - Quality is Essential.” In: 10th Anniversary International Workshop on Requirements Engineering: Foundation for Software Quality, REFSQ 2004, 2004.

[58] S. Jacobs, “Introducing measurable quality requirements: a case study,” In Proceedings 4th IEEE International Symposium on Requirements Engineering, pages 172–179, 1999.

[59] Floyd, C, “A Systematic Look at Prototyping,” In: Budde, R., Kuhlenkamp, K., Mathiassen, L. (eds.) Approaches to Prototyping, pp. 1–17. Springer, Berlin, 1984.

[60] E Hochmüller, “Towards the Proper Integration of Extra-Functional Requirements,” The Australian Journal of Information Systems 7, 98–117, 1999.

[61] Dieter Landes, "Requirements Engineering for Quality Requirements – Industrial Problem Statement", Fourth International Workshop on Requirements Engineering: Foundations of Software Quality REFSQ’98, Pisa, Italy, 1998.

[62] L. Chung, B.A. Nixon, E. Yu, and J. Mylopoulos, “NFR in Software Engineering,” Kluwer Academic Publishers, 2000.

[63] T. Gorschek and A. Davis, “Requirements engineering: In search of the dependent variables,” Information and Software Technology, 50:67–75, 2008.

[64] M. Lubars, C. Potts, and C. Richter, “A review of the state of the practice in requirements modeling,” In Proceedings 1st IEEE International Symposium on Requirements Engineering, pages 2–14, 1993.

[65] E. Kamsties, K. Hörnmann, and M. Schlich, “Requirements engineering in small and medium enterprises,” In International Proceedings Conference on European Industrial Requirements Engineering, pages 84–90, 1998.

[66] EASE project report 2009. Available at: http://ease.cs.lth.se/fileadmin/ease/2009/EASE_Activity_report_2009_v1_1.pdf

[67] Uusitalo, E.J., Komssi, M., Kauppinen, M., Davis, A.M., “Linking Requirements and Testing in Practice,” In: 16th IEEE International Requirements, 2008.

[68] Post, H., Sinz, C., Merz, F., Gorges, T., Kropf, T., “Linking Functional Requirements and Software Verification,” In: 17th IEEE International Requirements Engineering Conference, pp. 295–302. IEEE Computer Society, Atlanta, 2009.

[69] Sjoberg DIK, Anda B, Arisholm, Dybå T, Jørgensen M, Karahasanovic A, Espen F. Koren and Vokác M, “Conducting realistic experiments in software engineering,” In Proceedings of the 18th international symposium on empirical software engineering, pp 17–26, 2007.

[70] Hsia P, Davis AM, Kung DC, “Status report: requirements engineering,” IEEE Softw 10(6):75–79, 1993.

[71] Neill CJ, Laplante PA, “Requirements engineering: the state of the practice,” IEEE Softw 20(6):40–45, 2003.

[72] Potts C , “Software-engineering research revisited,” IEEE Software 10(5):19–28, 1993.

[73] Zelkowitz MV, Wallace, “Experimental validation in software engineering,” Information and Software Technology, 39(11):735–743, 1997.

[74] Maiden N, “What has requirements research ever done for us?,” IEEE Software 22(4):104–105, 2005.

[75] Pfleeger SL, “Understanding and improving technology transfer in software engineering,” Journal of Systems and Software, 47(2–3):111–124, 1999.

[76] Dyba, T.; Kitchenham, B.A.; Jorgensen, M., "Evidence-based software engineering for practitioners," *Software, IEEE* , vol.22, no.1, pp. 58- 65, Jan.-Feb. 2005

[77] Davis A, Dieste O, Hickey A, N Juristo, AM Moreno, “Effectiveness of requirements elicitation techniques: empirical results derived from a systematic review,” In: Proceedings of the 14th IEEE international requirements engineering conference (RE’06), IEEE Computer Society, pp 176-185, 2006/

[78] A. van Lamsweerde, “Requirements engineering in the year 00: A research perspective,” In Proceedings of the 2000 International Conference on Software Engineering, pages 5–19, 2000.

[77] Pfleeger SL, “Understanding and improving technology transfer in software engineering,” J Syst Softw 47(2–3):111–124, 1999.

[78] Gorschek T, Garre P, Larsson S and C Wohlin, “A model for technology transfer in practice,” IEEE Software, 23(6):88–95, 2006.

[79] Parviainen P and Tihinen M, “A survey of existing requirements engineering technologies and their coverage,” Int J Softw Eng Knowl Eng 17(6):827–850, 2007.

[80] Kitchenham B, Pearl Brereton O, Budgen D, Turner M, Bailey J, Linkman S, “Systematic literature reviews in software engineering—a systematic literature review,” Information and Software Technology, 51(1):7–15, 2009.

[81] B.A. Kitchenham, “Guidelines for performing systematic literature reviews in software engineering,” Tech. Rep., EBSE-2007-001, UK, July, 2007.

[82] Creswell, W. J., Research Design: Qualitative, Quantitative, and Mixed Method Approaches, Second Edition, Thousand Oaks: Sage, 2003.

[83] Dawson, C. W., The essence of computing projects a student's guide, Prentice Hall, Harlow, 2000.

[84] Robert K. Yin. Case study research: design and methods. Sage Publications, 3rd edition,
2003.

[85] P. Runeson and M. Höst, “Guidelines for conducting and reporting case study research in software engineering,” Empirical Software Engineering, 14(2):131–164, 2009.

[86] Claes Wohlin, Per Runeson, Martin H¨ost, Magnus C. Ohlsson, Bj¨orn Regnell, and
Anders Wesslen, “Experimentation in software engineering: an introduction,” (international
series in software engineering) Springer, 2000.

[87] T. Dingsøyr and T. Dyba, "Empirical studies of agile software development: A systematic review," Information and Software Technology, vol. 50, 2008, pp. 833-859.

[88] Seaman, C. B., “Qualitative methods in empirical studies of software engineering”, IEEE Trans. Softw. Eng, vol.25, no.4, 1999.

[89] Zeist, R.H.J. & Hendriks, P.R.H, “Specifying software quality with the extended ISO model,” Software Quality Management IV – Improving Quality, BCS, 145-160, 1996.

[90] Chen, P., Mead, N. R., Dean, M., Lopez, L., Ojoko-Adams, D., Osman, H. Xie, N., “SQUARE Methodology: Case Study on Asset Management System (CMU/SEI-2004-SR-015),” Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2004.

[91] C.Wholin, P. Runeson, M. Host, M.C. Ohlsson, B. Regnell and A. Wesslen, Experimentation in Software Engineering An Introduction, Kluwer Academic Publishers, Dordrecht, the Netherlands, 2000.

[92] J. Fleiss, “Measuring nominal scale agreement among many raters,” Psychological Bulletin, vol. 76, no. 5, pp. 378–382, 1971.

[93] Kitchenham, B.A. Pfleeger, S.L. Pickard, L.M. Jones, P.W.; Hoaglin, D.C. El Emam, K.; Rosenberg, J. , "Preliminary guidelines for empirical research in software engineering," *Software Engineering, IEEE Transactions on* , vol.28, no.8, pp. 721- 734, Aug 2002.

[94] B. Kitchenham and S. Pfleeger, “Software quality: the elusive target,” IEEE Software, vol. 13, no. 1, pp. 12–21, 1996.

[95] W. Scacchi, “Process models in software engineering,” Encyclopedia of Software Engineering, pp. 993–1005, 2001.

[96] R. van Solingen, D. F. Rico, and M. Zelkowitz, “Calculating software process improvement’s return on investment,” in Advances in Computers, 2006, vol. 66, pp. 1–41.

[97] Boehm, Barry and In, Hoh. “Identifying Quality-Requirement Conflicts”. IEEE Software, March 1996, pp. 25-35.

[98] Chung L., “Representing and Using Non-Functional Requirements: A Process Oriented Approach” Ph.D. Thesis, Dept. of Comp.. Science. University of Toronto, June 1993.

[99] Lauesen s. Software Requirements – Styles and Techniques. Addison–Wesley, 2002.

[100] I. Sommerville. Software Engineering. Addison–Wesley, 2007.

[101] R. Thayer and M. Dorfman, “Systems and Software Requirements Engineering,” IEEE Computer Society Press, 1990.

[102] Cabrero, D. Garzas and J, Piattini, M, "Choosing the Best Design Strategy from Requirements. A Value-Based Approach," *Exploring Quantifiable IT Yields, 2007. EQUITY '07. IEEE International Conference on* , vol., no., pp.87-94, 19-21 March 2007.

[103] Ebert, C, “Dealing with Nonfunctional in Large Software System”s. Annals of

Software Engineering, 3, 1997, pp. 367-395.

[104] Cysneiros, L.M. and Leite, J.C.S.P, “Integrating Non-Functional Requirements into data model” 4th International Symposium on Requirements Engineering – Ireland June 1999.

[105] Yu, Eric, “Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering” Proc. of the 3rd Interna. Symp. on Requirements Eng. Jan 1997 pp:226-235.

[106] Lindstrom D.R, “Five Ways to Destroy a Development Project” IEEE Software, September 1993, pp. 55-58.

[107] B.W. Boehm, J.R. Brown, and M. Lipow, “Quantitative evaluation of software quality,” In Proceedings 2nd International Conference on Software Engineering, pages 592–605, 1976.

[108] L. Chung, B.A. Nixon, E. Yu, and J. Mylopoulos. NFR in Software Engineering. Kluwer Academic Publishers, 2000.

[109] A. Borg, A. Yong, P. Carlshamre, and K. Sandahl, “The bad conscience of requirements engineering: An investigation in real–world treatment of non–functional requirements,” In Proceedings of the third Conference on Software Engineering and Practice in Sweden, pages 1–8, 2003.

[110] Madhavji NH, Holtje D, Hong W and Bruckhaus T, “Elicit: a method for eliciting process models,” In: Proceedings of CAS conference, Toronto, Canada, 31 October–3 November, pp.11-122, 1994.

[111] Ruhe G, Eberlein A, and Pfahl D, “Quantitative WinWin – A new method for decision support in requirements negotiation,” In: Proceedings of the 14th International Conference on Software Engineering and Knowledge Engineering (SEKE’02), ACM Press, New York, pp. 159-166, 2002.

[112] Kazman R, In H P and Hong-Mei Chen, "From requirements negotiation to software architecture decisions," Information and Software Technology, v 47, n 8, p 511-20, 1 June 2005.

[113] Korson, T. D. and Vaishnavi, V. K., “Managing emerging software technologies: a technology transfer framework,” *Commun. ACM* 35, 9, 101-111, 1992.

[114] Jones, C. , "Why is technology transfer so hard?," *Computer* , vol.28, no.6, pp.86-87, Jun,1995.

[115] Svahnberg M, Gorschek T, Feldt R, Torkar R, Saleem S.B, Shafique M.U, "A systematic review on strategic release planning models," Information and Software Technology, v 52, n 3, p 237-248, March 2010.

[116] Jørgensen M and Shepperd M, “A systematic review of software development cost estimation studies,” IEEE Transactions on Software Engineering, 31(1):33–53, 2007.

[117] Gorschek T, Garre P, Larsson S and Wohlin C, "Industry evaluation of the requirements abstraction model," Requir Eng 12(3):163–190, 2007.

Appendix A

Interview questions and description

1	**What is your name and designation (contact information)?**	Firstly, the work experience of the participant is identified. To find product domain specific QR practices, type of products developed and the company size
2	**What is the size of your company? Less than 50, 50-250 250-500, 500-1000, 1000-5000 or more than 5000**	

3	**What are your role and responsibilities in this organization?**	are identified. To capture the details of the department and persons responsibilities for handling QR, roles and responsibilities in brief.
4	**How many years of working experience you have in handling requirements/ requirements engineering?**	
5	**Please give a brief description of products and type of customers your organization deals with?**	
6	**Which department is responsible for requirements at your organization?**	
7	**Please briefly explain, which other persons are involved in handling requirements and their roles in your organization?**	

1	**Please describe the process of RE in your organization?**	To understand process of incorporating QR in general.
2	**What notion does the RE dept. have on quality requirements (non-functional)? Please describe** **In brief.** **[Very important to include in projects or products/ not so important/important/depends]**	To identify views/notions/approaches towards QR.
3	**What are the various types of quality requirements that you have successfully incorporated in projects?**	To identify various QR that are successfully incorporated, thereby establishing a sort of mapping of expertise and QR.
4	**How often are QR removed from the projects and why?** **Please mention the reasons of dismissal (time factor, difficulty in identifying value, cost factor, difficulty in finding right techniques etc).**	To know the views involved in the dismissal/removal of QR.
5	**Do you have any specific strategies, prioritization models in incorporating any specific types of QR or dismissing them?**	To identify how a list of QR are transformed into candidate QR for release/development.
6	**Do your customers give Requirements specification, or you develop your own? Specifically what does the procedure look like?**	To identify the source of requirements.
7	**Do you have any special techniques/models to specify quality (non-functional) requirements? If yes, please give a brief description. Please mention if you find any challenges in them.**	To know how QR are specified.

8	**What do you understand by "measurable quality requirements", "quantification of quality Requirements"?**	To identify a generalized view on what is meant by measurable QR.
9	**What measures do you collect for various QR?**	To know measures that are well pronounced in terms of practice.
10	**What do you understand by 'testable QR'? In relation to QR, measures, specification and testability are intertwined. Specific to this, what challenge do you find most common?**	To know viewpoint on achieving testable QR and identifying challenges.

Appendix B

Interview questions and answers (verbatim of responses)

B.1 Company alpha

Part I

1) What is your name and designation (contact information)?

Elli Tzatzani
Siemens Enterprise Communications
Requirements Engineer/Systems Analyst
Athens, Greece
Mob.: +30-6974554197
Skype: ellitzatzani

2) What is the size of your company?

Less than 50, 50-250 250-500, 500-1000, 1000-5000 or more than 5000

More than 5000 – actually more than 14000

3) What are your role and responsibilities in this organization?

Requirements Engineer/Systems Analyst

My responsibilities include analysis of Business requirements of customers, Product Management and new feature requests coming from in-house development, prioritization of their requirements depending on the business needs, risk analysis of the introduction of new components in our software and writing of user stories so that the requirements are understood from product management and development and finally cooperation with system test so that the quality of the requirements is assured.

4) How many years of working experience you have in handling requirements/ requirements engineering?

3 years

5) Please explain briefly about products and type of customers your organization deals with?

Products are Telephony Centers, SBCsm Phone devices, Media Servers, CTI Application servers, including the Management Applications.

Very large enterprises and organizations

6) Which department is responsible for requirements at your organization?

Product Management

7) Please briefly explain, which other persons are involved in handling requirements and their roles in your organization?

Stakeholders, requirements engineers, business analysts, systems analysts, project managers, product managers, service organization product managers, architects

Part II

1) Please describe the process of RE in your organization?

New requirements are provided from different teams/stakeholders.

RE department, participates in meetings with Product Management and Service Product Management where the validity and the added value of the new requirements is discussed. This team accepts or dismisses the new requirements.

For the accepted requirements: RE team analyses them, consults Software architects and proposes a solution.

Product management accepts the solution or rejects it.

If the solution is accepted, then Development estimates its effort.

Based on effort estimation, Product Management plans the version to deliver the requirement.

If the solution is rejected, the RE should come up with a new solution and so on until accepted.

2) What notion does the RE dept. have on quality requirements (non-functional)? Please describe briefly.

[Very important to include in projects or products/ not so important/important/depends]

Very important to include in products since this would avoid customer dissatisfaction at first place and will result to greater customer approval for the new products/releases.

3) What are the various types of quality requirements that you have successfully incorporated in projects?

Performance

Transaction time

Logging

Alarming

Installation time

Customer Training time

Time to configure system

Backup/Restore Time

Multiple-simultaneous-users support

Security

Traceability

System Uptime

Bulk changes performance

Multiple browsers support

Responsiveness to user actions

Customization

Error handling

Start-up time

Migration

Upgrades

Updates

4) How often QR are dismissed from the projects and why?

Due to time limitations and overload from errors in the implementation and customer complaints, each and every project dismisses the QRs. The time to deal with them comes after the product is delivered to the customer and after the customer complaints.

5) Do you have any specific strategies, prioritizing models and any trade-offs considered in incorporating any specific types of QR or dismissing them?

Prioritization takes place according to the customer business case and to the profit maximization or cost minimization that a requirement offers. The business case is calculated and according to the findings, the requirements are selected.

6) Do your customers give Requirements specification, or you develop your own? Specifically what does the procedure look like?

Customers are supposed to give requirements specifications. But, according to my experience, customers deliver a couple of words or phrases, usually complaints and RE & PM have to interview them for more details, to understand their problems and their business needs.

7) Do you have any special techniques/models to specify quality (non-functional) requirements? What are those? Do you find any challenges in them, i.e. understanding the specified QR?

No

a) **Please explain in brief how you specify QR so that they are understandable to designers? Do you use any techniques/methods/models like UML, or text based specification?**

Unfortunately we do not use a specific technique to describe QR. We use a couple of words as a description e.g. "System must support max 30 concurrent logins" OR "minimum number of users supported is 10" etc.

Maybe it is due to the nature of the SW that we implement, but, imo we need to find a better way to more accurately describe them.

b) **In this regard do you find a need for better ways of specifying QRs? If yes, specifically in what direction?**

We definitely need this, towards the direction of clearly stating which are the limitations so that they are not mis-interpreted from development

6) Do your customers give Requirements specification, or you develop your own? Specifically what does the procedure look like?

Customers are supposed to give requirements specifications. But, according to my experience, customers deliver a couple of words or phrases, usually complaints and RE & PM have to interview them for more details, to understand their problems and their business needs.

7) Do you have any special techniques/models to specify quality (non-functional) requirements? What are those? Do you find any challenges in them, i.e. understanding the specified QR?

No

a) **Please explain in brief how you specify QR so that they are understandable to designers? Do you use any techniques/methods/models like UML, or text based specification?**

Unfortunately we do not use a specific technique to describe QR. We use a couple of words as a description e.g. "System must support max 30 concurrent logins" OR "minimum number of users supported is 10" etc.

Maybe it is due to the nature of the SW that we implement, but, imo we need to find a better way to more accurately describe them.

b) In this regard do you find a need for better ways of specifying QRs? If yes, specifically in what direction?

We definitely need this, towards the direction of clearly stating which are the limitations so that they are not mis-interpreted from development

8) What do you understand by 'measurable quality requirements', 'quantification of quality requirements'?

'measureable': measure the functionality of each one and assess the impact of including it.

'quantification': quantify the cost of success/failure to include this requirement in a solution

9) What measures do you collect for various QR?

Nothing

10) What do you understand by 'testable QR'? In relation to QR, measures, specification and testability are intertwined. Specific to this, what challenge do you find most common?

'testable' QR is a QR that is described adequately enough so that is provides the appropriate guidelines to testers to test this effectively.

Most common challenge is to provide a performance index for the new requirements.

11) Are the testing techniques implemented at your company well suited to test QRs? In this regard do you find a need for knowledge transfer of better techniques/ways for testing QRs?

No, it is done manually and using some free-ware tools. We definitely need this knowledge.

B.2 Company beta

1	**What is your name and designation (contact information) ?**	Trond Johansen Trond.Johansen@proxycom.no
2	**What is the size of your company? Less than 50, 50-250 250-500, 500-1000, 1000-5000 or more than 5000**	Less than 50
3	**What are your role and responsibilities in this organization?**	Systems development manager
4	**How many years of working experience you have in handling requirements/ requirements engineering?**	30 years
5	**Please give a brief description of products and type of customers your organization deals with?**	Development of web applications for customers
6	**Which department is responsible for requirements at your organization?**	The system development department
7	**Please briefly explain, which other persons are involved in handling requirements and their roles in your organization?**	3 System Developers

1	**Please describe the process of RE in your organization?**	The requirements specifications are developed in cooperation with customer representatives according to a simpler variant of the Rational Unified Process (RUP).
2	**What notion does the RE dept. have on quality requirements (non-functional)? Please describe** **In brief.** **[Very important to include in projects or products/ not so important/important/depends]**	Some of the non-functional quality requirements are very important to include in the requirement specification.
3	**What are the various types of quality requirements that you have successfully incorporated in projects?**	We have mostly used: Accessibility, Integrity, Presentability, Usability, Timeliness, Future Business Potential.
4	**How often are QR removed from the projects and why?** **Please mention the reasons of dismissal (time factor, difficulty in identifying value, cost factor, difficulty in finding right techniques etc).**	It depends on the project. However, some of the QR in 3 are always present. The reason of dismissal are: - The QR is not of importance in the project
5	**Do you have any specific strategies, prioritizing models in incorporating any specific types of QR or dismissing them?**	The requirement specification is developed in cooperation with the customers.

6	**Do your customers give Requirements specification, or you develop your own? Specifically what does the procedure look like?**	The requirement specification is developed in cooperation with the customers.
	In cooperation with the customer do you use checklist to select QRS? Are the checklists by understanding quality models like ISO? Are the checklists tailor made for the projects? Please describe the process in brief.	We use a checklist based on the ISO standard. The checklist is used in all projects, and the QR are specified in cooperation with the customers.
7	**Do you have any special techniques/models to specify quality (non-functional) requirements? If yes, please give a brief description. Please mention if you find any challenges in them.**	The QR specification is text based.
	Do you find any challenges in the text based specifications? **Do you find techniques like UML [way of specifying/modelling] QR difficult/not suitable for your projects? Were there any knowledge transfer efforts in this regard in the past?**	The use of text based QR specification is OK. The use of modelling techniques like UML is not suitable.
8	**What do you understand by “measurable quality requirements”, “quantification of quality Requirements”?**	To give a value to a quality requirement.
	Do you find any challenges in assigning/identifying values to QR? Do you use any models? What are the most value generating components in the projects?	Yes, some QR may be difficult to specify, like: Maximum data base size, maximum transactions/second, etc. The GUI and the data base components are usually the most value generating components.
9	**What measures do you collect for various QR?**	Examples: No of errors; Mean time between errors; and so on.
10	**What do you understand by ‘testable QR’? In relation to QR, measures, specification and testability are intertwined. Specific to this, what challenge do you find most common?**	That QR can be measured such as: processing time, MTBF. Challenge: Avoid the QR term, and use more specific terms like: Response time, processing time, etc, instead of Efficiency
	How do you validate you have successfully incorporated the right amount of quality [requirements] in your projects? Can you successfully ensure this through the use of testing techniques (like threat trees, misuse cases etc.)? Please mention some of the techniques you find helpful.	The validation is performed by other system developers, and any comments are discussed with the customers and may be accepted.
11	**How often the following situations pop up/appear in your projects?** **a) Ah, it is too late to include a particular QR.** **b) We did not prioritize QRS for implementation and now they are backfiring [because of time and cost factor]** **c) We have to implement a few QRs but we are no so good or not sure about testing them [and**	a) Never too late. A QR can be included, but may have cost and time consequences for the project. b) Same as a. c) May happen in a few projects.

identifying if the required level of quality is present]

B.3 Company gamma

Verbatim of the response: We do handle requirements but not in a way that I feel I can answer your questions. We always get our requirements from an RFQ “Request for Proposal”. These requirements are non-negotiable. So in basic we state compliant or non-compliant to the RFQ. We don’t have the possibility to influence the requirements.