

*Master Thesis*  
*Software Engineering*  
*Thesis no: MSE-2001-13*  
*October 2001*



# **Trade-offs and Conflicts between Quality Attributes**

**A Study of Academia and Industry**

**Kennet Henningsson**

Department of  
Software Engineering and Computer Science  
Blekinge Institute of Technology  
Box 520  
SE – 372 25 Ronneby

This thesis is submitted to the Department of Software Engineering and Computer Science at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Software Engineering. The thesis is equivalent to 10 weeks of full time studies.

**Contact Information:**

Author:

Kennet Henningsson

Address:

IPD

Blekinge Institute of Technology

Box 520

372 25 RONNEBY

E-mail: [Kennet.Henningsson@bth.se](mailto:Kennet.Henningsson@bth.se)

University advisor:

Claes Wohlin

Department of Software Engineering and Computer Science

Department of  
Software Engineering and Computer Science  
Blekinge Institute of Technology  
Box 520  
SE – 372 25 Ronneby

Internet : [www.bth.se/ipd](http://www.bth.se/ipd)  
Phone : +46 457 38 50 00  
Fax : + 46 457 271 25

# ABSTRACT

It is next to impossible to let a day go by without coming into contact with a computer system in some way, either by direct usage, or making a telephone call that is directed by computer systems. Due to the importance of computer systems we are all concerned with their quality, directly or indirectly, a malfunctioning system will disturb our lives in more or less drastic ways. Each stakeholder promotes his qualities and priorities for the system, and this will eventually lead to conflict, depending on relations between qualities. There is no simple solution to the problem of relating and conflicting quality attributes. But by making an informed decision it is possible to soften the effects of the trade-offs and gain a higher level of total software quality. The way to make an informed decision is to have knowledge about the relations between quality attributes along with insights of the consequences of the decision. This report gives a short introduction to the problem, and surveys both the relations stated within academia, and industry. The intention is to explore the knowledge present within academia and collect and present the established relations found by researchers. Further this report presents a survey carried out at companies working in different areas with software engineering, to show which relations that are visible to industry and how they are handling the relations between quality attributes. This gathered knowledge intends to support the ambition towards higher quality in software engineering.

**Keywords:** Quality attributes relations, stakeholder, quality attribute trade-offs, prioritization between qualities.

# CONTENTS

<b>ABSTRACT .....</b>	<b>I</b>
<b>CONTENTS .....</b>	<b>II</b>
<b>1 INTRODUCTION .....</b>	<b>1</b>
<b>2 BACKGROUND.....</b>	<b>2</b>
2.1 PROBLEM DESCRIPTION .....	2
<b>3 QUALITY ATTRIBUTES IN A CONTEXT .....</b>	<b>4</b>
3.1 SOFTWARE ENGINEERING AND QUALITY ATTRIBUTES.....	4
3.1.1 <i>Phases</i> .....	4
3.1.2 <i>System dynamics</i> .....	4
3.1.3 <i>System life cycle</i> .....	5
<b>4 RESEARCH QUESTION .....</b>	<b>7</b>
4.1 HYPOTHESES .....	7
<b>5 METHODOLOGY .....</b>	<b>8</b>
5.1 LITERATURE SURVEY.....	8
5.1.1 <i>Method</i> .....	8
5.1.2 <i>Sample</i> .....	8
5.2 INDUSTRY SURVEY .....	9
5.2.1 <i>Method</i> .....	9
5.2.2 <i>Sample</i> .....	9
5.2.3 <i>Line of questioning</i> .....	10
5.2.4 <i>Interview</i> .....	10
<b>6 RESEARCH.....</b>	<b>11</b>
6.1 INTRODUCTION .....	11
6.2 RESULT.....	11
6.2.1 <i>Quality attributes</i> .....	11
6.2.2 <i>Collection or relations</i> .....	13
6.2.3 <i>Handling the conflict</i> .....	14
6.3 STATE OF THE ART .....	15
6.3.1 <i>The conflict</i> .....	15
6.3.2 <i>Finding relations</i> .....	16
<b>7 PRACTICE .....</b>	<b>17</b>
7.1 INTRODUCTION .....	17
7.2 RESULT.....	17
7.2.1 <i>Quality attributes used</i> .....	17
7.2.2 <i>Measurement of quality attributes</i> .....	18
7.2.3 <i>Relations found in industry</i> .....	19
7.3 STATE OF THE PRACTICE .....	21
<b>8 ART VS. PRACTICE.....</b>	<b>22</b>
<b>9 CONCLUSIONS.....</b>	<b>23</b>
<b>10 REFERENCES.....</b>	<b>25</b>

# 1 INTRODUCTION

This thesis focuses on the relations between quality attributes, and the problems accompanied. A problem when developing software is to satisfy customers. However, it is not always the customers that use the software in the end. Moreover it is not always clear which effects the software will have on other people in private or in their business role. A conclusion is that there is a vast number of stakeholders (persons with an interest in the system) to each system, directly or indirectly. Achieving approval from all system stakeholders, and get a joint view of the relations for that system is hard to do (Boehm and In, 1996). This means that it is hard to reach a given level of quality for one or more specific quality attributes without decreasing the quality level for conflicting or non-stringent quality attributes.

This thesis takes the approach of improving and securing quality by better understanding through collecting knowledge about quality attribute relations within academia and industry. The knowledge could be used to avoid conflicts and to present a solution with prioritized quality requirements.

The method for extracting information from academia and industry is the survey method. A literature survey is used for academia, and structured interviews are used for acquiring knowledge from industry representatives.

The information gathered is presented in two separate sections with the intention to describe the state of the art and the state of the practice. The state of the art section presents academia's opinion about which quality attributes shall be used in order to secure high quality software and the relations between the quality attributes. The state of the practice describes how the quality issues are handled within industry, in the perspective of usage of quality attributes and how their relations are handled and monitored.

Conclusions do neither contain any magic stroke on how to resolve the conflicts nor how to increase the total quality, but recommendations on how to avoid the pitfalls and provide input to the prioritization process. So that achievable quality goals are produced which will, hopefully, lead to an increased overall quality.

It is suitable to establish a common terminology. There are three main types of terms describing roughly the same thing, they are:

- Quality factors, stated by McCall (1994)
- Non-functional requirements, stated in Chung, et al. (1999).
- Quality attributes, mentioned and used in Bosch (2000).

Throughout this report these terms will be used synonymously, depending which is most suited.

An important motivation for this thesis work is a well-founded interest, belief and importance, in developing software with a good balance between quality attributes, which in the same way fulfills its functional requirements and expectations.

There has been a great deal of work done, and still are, concerning each unique, or possible two or three, quality attributes. The problem arises when several quality attributes need to be maximized at the same time. This scenario can cause conflicts and prioritization between quality attributes is needed and in the end when prioritizing among quality attributes you is in some sense prioritizing the interests of stakeholders.

The outline for this report is as follows, Section 2 and 3 deals with background and context for quality attributes, while Section 4 and 5 handles the research question and method used. In Section 6 the findings from research are stated and Section 7 contain findings from industry. In section 8 the findings from academia and industry are related to each other, and Section 9 states the conclusions. The references used are listed in section 10.

## 2 BACKGROUND

The term quality has been used for some time; the first occurrence of the word is dated back to the 14<sup>th</sup> century according to Merriam-Webster's Collegiate Dictionary. The association of software quality with quality factors appeared in the late 1970s in conjunction with research and development of software measurement technology. According to McCall (1994) the software measurement (also known as metrics, or software metrics) evolution created the tools for measuring software quality.

A quote from Albert Einstein: "Not everything that counts can be counted; and not everything that can be counted counts". This is found to be true, when dealing with quality attributes as well, even if you have a measure, it is not by default the measure that represents quality, and if it is not suited for measuring, it is no indication that it is unimportant for software quality.

Over time many authors and researchers have created a number of new, or some more or less modified quality attributes, these are shown in table 1.

McCall, 1977	Boehm, 1978	Bowen, 1985	Murine, 1983	Others*
Correctness	Reliability	Correctness	Correctness	Correctness
Reliability	Efficiency	Reliability	Reliability	Reliability
Efficiency	Human Engineering	Efficiency	Efficiency	Efficiency
Integrity	Understandability	Usability	Usability	Usability
Maintainability	Modifiability	Integrity	Integrity	Integrity
Flexibility	Testability	Maintainability	Maintainability	Maintainability
Testability	Portability	Flexibility	Flexibility	Flexibility
Portability		Verifiability	Testability	Testability
Reusability		Portability	Portability	Portability
Interoperability		Reusability	Reusability	Reusability
		Interoperability	Interoperability	Interoperability
		Survivability	Intraoperability	Survivability
		Expandability		Safety
				Manageability
				Functionality
				Supportability

Table 1: Authors and quality attributes.

### 2.1 Problem description

To any modern software system there is a number of stakeholders. The term stakeholder can be explained as follows: "System stakeholders are people or organizations who will be affected by the system and who have a direct or indirect influence on the system requirements." (Kotonya and Sommerville, 1997).

As will be described further in Section 3. The software engineering process consists of different phases, requirements gathering, analysis, architecture development, design, implementation, testing, verification and validation and so on. Additionally, when the system is delivered to the users or organization, the maintenance and preserving phase will be initiated. The point is, through these phases' different requirements or qualities will be promoted, and required by different stakeholders.

It is not an easy task to make the right prioritizations among the quality attributes and looking ahead into the remaining lifetime of the system. If the relations among the

---

\* Others are: Grady and Carswell (1987); Deutsch and Willis (1988); Evans and Marciniak (1985); Arthur (1985).

quality attributes are unknown to you, it is easy to make the wrong decision, and falling for the momentarily most important attribute, and by that create problems further on. By knowing and making the relations and influences between quality attributes visible, it ought to be easier to make the right decision and simplify the fulfillment of non-functional requirements throughout the systems lifecycle.

There is a need for knowledge about the processes of software engineering as well, since the quality attributes and their relations are not an isolated occurrence within the discipline. Quality attributes, non-functional requirements and quality goals are set and interpreted based on their context, which is why it is necessary to describe the context of quality attributes.

### 3 QUALITY ATTRIBUTES IN A CONTEXT

It is not suitable to introduce the usage and relations of quality attributes without putting the quality attributes in a context. It is necessary to have some understanding of the software engineering process and the interests/demands placed on software. The term quality attribute is perhaps not unknown to people within software engineering, but history has also shown that there are many different interpretations.

#### 3.1 Software engineering and quality attributes

Software engineering, from the viewpoint taken in this thesis report, is focused on requirements fulfillment placed on the final system from different stakeholders with different priorities. Stakeholders and their priorities may alter from different phases in the systems life cycle, from phases in the development process, and depending on the system's dynamics as well.

##### 3.1.1 Phases

When developing software there are some basic steps that need to be taken. You must know what you are supposed to do, you must know how well you are supposed to do it, and finally what will happen to the system after it has been delivered.

There is a selection of models present which have their own set of phases and motives for why exactly that model should be used, some examples are:

- Waterfall model, which consists of sequential steps carried out in order. The result from one step is the needed input to the next step. (Pfleeger, 2001)
- Spiral model, where iterations are introduced. Each step is carried out in each iteration and thus building the knowledge over time. (Boehm, 1988)
- Evolutionary Model, which develops and delivers subsets of the total system over time, which in the end will lead to a total system delivered. (Gilb, 1997)

Even though these, and other, development models are different, substantially different in some cases, they contain roughly the same setup of phases, viewed from the total lifecycle for the software system. The development models contain the generic phases: requirements collection, architecture/analysis/design, implementation, verification and validation where testing is included, launching, operation and maintenance.

Through these phases, the stakeholders are involved more or less, in this sense the stakeholders represent interests for the final system, the same person can be holding different interests and may thereby be viewed as different stakeholders. Stakeholders, by definition ("direct or indirect influence on the system requirements") set the requirements for the system (Kotonya and Sommerville, 1997). The requirements are both functional and non-functional requirements. The non-functional requirements can be formulated as quality goals and expressed by using quality attributes, so it is possible to say that stakeholders have their own set of quality attributes that they are prioritizing.

##### 3.1.2 System dynamics

When the software system is developed the code is static and viewable by reading it in a text editor. This changes when the system executes, when the system is operational it is dynamic, especially in object-based programs it is hard to control the number of objects that exists and in which state the single objects are. In this state the system is dynamic, and quickly changeable.

When the system is developed, when the code is written, the system is static, even here there are a number of stakeholders. An example would be the testers or the testing organization, that has an interest in that the original code is developed with high



testability, and further the business and sales person would be interested in how fast the system is developed.

When the system is in operation, users and system administrators are possible stakeholders and could then be prioritizing quality attributes such as efficiency, flexibility and integrity.

During maintenance of a system, the system can be both static and dynamic. It is not sure that all parts of the software system are brought down in order to maintain certain parts. Then there are stakeholders interested in both the dynamic aspect and the static aspects, there is also separate setup or usage of quality attributes, depending on if the static or dynamic behavior is described. The maintenance organization is interested in maintainability of the system, and users are interested in keeping the maintenance downtime as short as possible. Further interested parties may be testers that are interested in testability and integration of the module updated through maintenance and how it shall be possible to test this in collaboration with the running system.

### 3.1.3 System life cycle

A wider perspective on phases and of the system is to view the whole lifecycle from initiation to retirement of the system.

The initiation period is conducted with help of customers, end-users and development representatives, these persons are in this case stakeholders, and they have varying objectives and priorities for the system. An example would be that the customer who is purchasing the system could be interested in a low development cost, while the end-user of the system is interested in maximum usability which may increase development cost. This is accompanied by the developing organization who can be interested in making a general solution in order to reuse the system, or parts there of. The interest in the system may vary, and can cause conflicts on to which requirements to be laid onto the system, both functional and non-functional.

During the development phase of the lifecycle, another set of stakeholders and requirements are introduced onto the system, here the importance will be to develop a stable and efficient system, which fulfills the user requirements, functional and non-functional. In parallel with the user requirements, developers are faced with requirements from maintenance and testing organizations, about testable, and maintainable code. Another addition can be the sales or financial aspects, the contract may be on fixed price and so the development costs shall be held to a minimum to secure profit.

When the maintenance phase of the cycle is introduced, stakeholders are promoting requirements such as maintainable systems along with short lead time for maintenance and high reliability after maintenance which implies testability as well. During maintenance interests about portability may surface, in the sense of porting the system to another hardware or software platform.

The next, and ending step, in a systems life cycle is retirement. When a system is brought down from operation, it is often replaced by another system. What is important during this event is portability of data, and perhaps interoperability, so that it is possible to export the information from the retired system into the new. It is not desirable to loose any information, when the system is retired and the files and databases are deleted. There may also be legal aspects for how long the information must be stored and available.

The problem here is to look far ahead, facilitating changes and modifications likely to happen down the line. There is a reluctance against prioritizing potential changes before the real problems which you are faced with at the present time, even though research shows that a great deal of the effort and cost are spent in maintenance through the systems lifecycle. Bosch (2000) mentions figures as high as 80 percent spent on maintenance in his book *Design & Use of Software Architecture*. Based on this it should be prioritized to ease the maintenance effort by prioritizing maintenance when developing software.

The various priorities and requirements and the conflicts between quality attributes will lead to problems if you are not aware of the conflicts. If the conflicts are acknowledged, it is possible to make correct decisions and thereby acquiring high and lasting quality for the software system developed.

This leads on to the research question, which aims to find potential relations and conflicts described in literature and within industry.

## 4 RESEARCH QUESTION

The research question is “What are the characteristics of a set of quality attributes, which quality attributes does it contain, what are their relations, and what are the conflicts?”

The wording “set of quality attributes” is chosen depending on the explanation of sets. “A set is what mathematicians call a collection of objects - any objects - when the order of the objects is irrelevant.” (Maurer and Ralston, 1991).

The research question is applicable both for research and an industry perspective. It is also possible to break down the all-embracing question into smaller parts as well.

Viewing the research question from an industry perspective the question would focus on which quality attributes that are used within the organization at hand, and how the organization perceives the relations between the used quality attributes.

When looking upon the research question from a research perspective, the question will focus upon which quality attributes are present in the literature as well as how the research community perceives the relations between quality attributes.

The research question, the outline and handling of this thesis work are influenced by experiences and hypotheses as well. My experience is based on personal participation in student projects as well as some short time employment during education. The interest for quality is based on the assumption that “we can do better than this”, true or not.

From this point of view the research question was formulated and hypotheses were created.

### 4.1 Hypotheses

The formulation of the survey questions for industry as well as for the extraction of information from the research community is based on some basic hypotheses which are:

- Industry only uses a subset of the quality attributes described in literature. Not all quality attributes are taken into consideration within the process.
- All quality attributes are not measured. Some are measured quantitatively, some are measured qualitatively, and some are not measured at all.
- The attributes that are used in the software engineering process are also controlled in some way, not necessarily through quantitative measurement.
- There exist quality attributes within industry that are not described in literature, depending on the marketing demands set on the industry. These attributes are not necessarily mentioned as quality attributes. But never the less these are factors that influence the development of software.
- There exist relations among quality attributes that are not recognized in the requirements phase of software engineering projects.
- There exist relations that are conflicting as well as there exist relations that are stringent.

The hypotheses are more related to the usage of quality attributes within industry rather than to the exploration of knowledge present in the research community. Past experience within industry has led to these hypotheses, but the experience of what to expect from research and researchers are more limited.

The nature of the research question and the hypotheses influenced the selection of research method.

## **5 METHODOLOGY**

To answer the research question and to confirm or reject the hypotheses the research method survey was chosen and judged as suitable. The survey methodology can be described as collection of standardized information from a specific population, or some sample of one, usually but not necessarily by means of questionnaire or interview (Robson, 1998).

As the research question applies to two different parts, industry and academia, the research methodology must also be conformed onto these two samples. A structured interview was used for surveying the industry and a literature survey was used for surveying academia.

### **5.1 Literature survey**

The motivation for the literature survey is to expand the body of knowledge by gathering and in some cases comparing research material and findings. The reason for only using written material is that it is stable, the source material will be present and not contaminated after the survey has been conducted. The motivation for only using published material is based on that, published material has been reviewed, which to some extent assures quality.

#### **5.1.1 Method**

The process consists of four steps. These steps were only adapted to the searching for information in articles and papers.

- Keyword search. This is done by feeding a set of keywords into a number of search engines. The search engines are accounted for in the sample section below. The keywords were searched for in, where facilitated by the search engine, title, abstract, keyword list, and full text.
- Abstract analysis. This is done by reading the abstracts available, then the contents of the abstract were analyzed momentarily to judge if it reflected occasional occurrences of the keywords searched for or if the article handled the topic of interest.
- Shallow reading. This is done by browsing the article and focus on sections that contains valuable and related information for the subject of this thesis. The author of this report subjectively decides the selection of what seemed as related information, but hopefully, the selection was carried out roughly in the same way for all articles. Even though there is a risk for that implicit criteria change along with expanding knowledge gathered by reading the articles.
- Highlighting and short summary notes. This is done by highlighting sections and noting conclusions in the margin, or in digital notes to the report while reading the report. This step is often integrated with the prior step: shallow reading. When the highlighting is done a short summary is written and attached to the article, which summarizes the intention and subject of the report and the most prioritized or interesting findings within the article.

When dealing with books as a sources for information, the same methodology was used, roughly, if the book was judged as interesting, the contents listening was browsed through and the most suitable chapters or sections were summarized with page directions into where the quote or information was found.

#### **5.1.2 Sample**

The choice made to only use published material is based on the review process the published material has to pass. The intention by this decision is to assure that the

material found should have high quality, or at least such quality as the reviewing process assures.

The published material may be present in: journals, magazines, licentiate or doctoral thesis, or conference papers. The sample has not been limited to include or exclude any certain type of the mentioned publications. The search for publications was done by using search engines available at BTH (Blekinge Tekniska Högskola) library, Infocenter ([www.bth.se/bibliotek](http://www.bth.se/bibliotek)) and search engines at ACM ([www.acm.com/dl](http://www.acm.com/dl)) and IEEE digital library ([www.computer.org/dlsearch.htm](http://www.computer.org/dlsearch.htm)).

The sample of books was generally gathered the same way primarily by searching Infocenter for suitable books.

## **5.2 Industry survey**

There is a great deal of valuable knowledge within software engineering industry; this knowledge also covers quality attributes, their relations and possible conflicts. The knowledge is somewhat more practical and concrete than what is expressed within academia and written in publications. Industrial knowledge may not always be explicit and expressed in the terminology used within research. Though industrial knowledge is not less valuable or accurate.

### **5.2.1 Method**

The method used for surveying the industrial usage of quality attributes, their relations and conflicts are structured interviews (Robson, 1998). The structured interview facilitates for interviewees answering the same questions, but makes it possible for the interviewee to expand and elaborate without having to stick to a defined set of answers. The possibility for having a dialogue is desirable, when the questions can be explained and clarified if the questions are misunderstood. There may be a thin line between clarifying a question and making a leading question or directing the answer, this was observed and caution was taken.

The interview was recorded, naturally with consent of the interviewee. Afterwards the interview was transcribed, though not word by word, instead the answers were summarised and clarified, without altering the answer. If the interviewee was interested he or she received a copy of the transcript for approval or corrections of misunderstandings.

### **5.2.2 Sample**

The basic sample was created from the companies participating in the research profile application, where Blekinge Institute of Technology (BTH, the Swedish abbreviation) will collaborate with several companies.

Some of the participating companies in the profile application are:

- Ericsson Software Technology (Ericsson, 2001): Ericsson develops operational systems for telecommunications and mobile systems.
- Symbian (Symbian, 2001): Symbian develops operating systems and applications for wireless and handheld units primarily.
- Micronet (Micronet, 2001): Micronet develops applications and products for the Internet and mobile community, both by partners and in-house. They also supply consulting services.
- Europolitan (Europolitan, 2001): Europolitan develops administrative systems for their own organization as well as mobile services for the end customer using mobile devices within their mobile net.
- Teleca (Teleca, 2001): Teleca develops systems and services for most of the areas within software engineering, both wireless, integrated systems and drivers and hardware connected solutions. They also supply consulting services.

Since the interest is in quality attributes and there possible relations, the sample consists of persons in roles within the company that are likely to have knowledge or

experience concerning these possible relations. The persons interviewed had the roles of:

- Department manager.
- Quality responsible.
- Branch manager.
- Technical project manager.
- Product manager.

The level of theoretical education within the sample varies between senior high school degree and licentiate degree. The average study points achieved at university level are 146 study points. This sample gives insight and knowledge about quality attributes used and their potential relations and conflicts. Since the line of questioning is not focused on a low or explicit level, a broad knowledge about quality and prioritization of the organizations projects are enough to answer the questions accurately with satisfying credibility.

### 5.2.3 Line of questioning

The interview has three major phases, and two minor. The structure of warm-up and cool-off periods were applied, which means that the interview starts and ends with some easy and not crucial questions. The first phase was concerned with the interviewee's education, prior working experiences and the experience of the current position. The last phase was concerned with reflections of the subject and questions asked, and with openings to add further comments.

The three major phases are, in order, usage of quality attribute, measuring quality attributes or quality goals, and the relations/conflict between the quality attributes used.

### 5.2.4 Interview

The initial contact with the potential interviewees was taken through e-mail, which was written by the author and his supervisor, Claes Wohlin, and was sent to contact persons within the profile project. In order to book the interviews contacts through telephone were taken.

The interview was carried out in the office or conference room at the interviewee's company, this was done to shorten the time spent for persons participating in the interview. The time frame planned for the interview was 60 minutes, which seemed possible for interviewees to spare, along with recommendations given by Robson, saying that an interview should not be shorter than 30 minutes and not longer than 60 minutes (Robson, 1998).

As mentioned above the interviews were recorded, with interviewee consent, and a transcript was written from each interview and approved by the interviewee, if he or she found it necessary to do so.

Detailed information about the interview design is found in appendix A.

The results from the survey both within academia and industry are stated in the coming sections 6 Research, and 7 Practice.

## **6 RESEARCH**

The research material that is used in this master thesis is varying greatly in time, there are material that originates from the late 70's and onwards to 2001, which are the publishing date for some books used. The vast amount of research made during this time period is impossible to cover within the time frame given for this thesis, even with help of search engines and computers. So this section represents a sample of the research available.

### **6.1 Introduction**

Software quality is a subject that is often discussed within research and literature; there are almost as many opinions on how to achieve good quality, as there are authors. Though the opinions and solutions vary, the relations between software quality attributes are acknowledged by many authors, and almost naturally discharged by others. Some quotes with references are given:

“Finding the right balance of quality-attribute requirements is an important step in achieving successful software requirements and products. To do this, you must identify the conflicts among the desired quality attributes and work out a balance of attribute satisfaction.” (Boehm and In, 1996).

“... This can help reduce conflict among differing quality requirements. Whereas it may be impossible to deliver extremely high operational requirements in many different and sometimes conflicting dimensions for a product as a whole, it may be feasible to deliver stringent quality requirements for the individual product portions that genuinely require them.” (Bøegh et al., 1999).

“The challenge in software design is to find solutions that balance and optimize the quality attributes of the system. ... Maintainability and performance often serve as examples of inherently conflicting quality requirements where a trade-off is inevitable. ... An alternative design was implemented and evaluated. The evaluation results show that the alternative parser has much better performance characteristics as well as higher maintainability. These findings show that the design decisions were based on a general assumption that proved invalid, i.e. the performance problems were caused by a myth.”(Häggander et al., 1999).

These were some statements from research, which illustrates the problem and the fact that it does have more than one solution. One approach to the various opinions is that the research seldom is in complete agreement, which is the easy explanation. Another way to look at it would be to examine the granularity of the studies. The first two quotes came from general observations on relations among quality attributes. The third quote came from a case study based on a specific system, and a specific part of that system as well. This could partly explain why the opinions vary.

The intention of this report is to focus and describe the conflicts and trade-offs between quality attributes.

### **6.2 Result**

Since the problem of varying opinions is easy to acknowledge it is decided that, it is not useful to present yet another opinion. So the approach within this report will be to compile existing research and presenting it in an un-biased way. This is the result of the literature survey of research.

To further understand the meaning of quality attributes and be able to build an opinion, the definitions for quality attributes used will help.

#### **6.2.1 Quality attributes**

There is a number of quality attributes available in literature, which poses a problem as well. But potential misunderstandings of the meaning of a quality attribute

pose a more serious problem. This is why the definitions of quality attributes are of importance. The definitions offer a possibility to compare different definitions and build an opinion of which definition is most suitable for the organization at hand. This report states two definitions from McCall and ISO (International Organization for Standardization)

McCall used the following definitions for his quality attributes.

Quality factor	Definition
Correctness	Extent to which a program satisfies its specifications and fulfills the user's mission objectives.
Reliability	Extent to which a program can be expected to perform its intended function with required precision.
Efficiency	The amount of computing resources required by a program to perform a function.
Usability	Effort required to learn, operate, pre-prepare input, and interpret output of a program.
Integrity	Extent to which access to software or data by unauthorized persons can be controlled.
Maintainability	Effort required locating and correcting an error in an operational program.
Flexibility	Effort required modifying an operational program.
Testability	Effort required to test a program to ensure it performs its intended function.
Portability	Effort required transferring a program from one hardware configuration and/or software system environment to another.
Reusability	Extent to which a program can be used in other applications-related to the packaging and scope of the functions that programs perform.
Interoperability	Effort required coupling one system with another.

Table 2: McCall (1994) definitions of quality factors.

Definition of quality attributes given by ISO.

Quality factor	Definition
Functionality	The capability of the software product to provide functions, which meet, stated and implied needs when the software is used under specified conditions.
Reliability	The capability of the software product to maintain a specified level of performance when used under specified conditions.
Usability	The capability of the software product to be understood learned, used and attractive to the user, when used under specified conditions.
Efficiency	The capability of the software product to provide appropriate performance, relative to the amount of resources used, under stated conditions.
Maintainability	The capability of the software product to be modified. Modifications may include corrections, improvements or adaptation of the software to changes in environment, and in requirements and functional specifications.
Portability	The capability of the software product to be transferred from one environment to another.

Table 4: ISO (2000) definition of quality attributes.

There are some striking similarities among the different definitions, for example Efficiency in ISO and McCall. These similarities are useful to have in mind when material concerning quality and quality attributes are read, the names may differ, and



also the wording of the definitions, but when the abstraction is raised just a bit, the similarities are striking.

It is likely to believe that there are some generic and common qualities when dealing with software, and these common qualities cause problems from time to time. The conclusion that there is some common ground between the different definitions of the quality attributes is based on the similarities between the definitions.

### 6.2.2 Collection or relations

An important factor when discussing quality attributes is to know what the quality attributes actually mean. As table 1 shows there are a number of quality attributes and the fact that they have the same name is not a guarantee for the same meaning.

When examining the literature, authors have mentioned relating and conflicting quality attributes, which are stated below.

Quality attribute	vs.	Influence	Reference	Comment
Performance		Evolvability	Negative	Boehm and In 1996 No reference to definitions for the quality attributes was given.
Portability		Performance	Negative	
Assurance		Interoperability	Positive	
Assurance		Usability	Positive	
Assurance		Cost/Schedule	Negative	
Assurance		Performance	Negative	
Assurance		Evolvability	Negative	
Assurance		Usability	Negative	
Interoperability		Assurance	Positive	
Interoperability		Usability	Positive	
Interoperability		Cost/Schedule	Negative	
Interoperability		Performance	Negative	
Evolvability/Portability		Interoperability	Positive	
Evolvability/Portability		Reusability	Positive	
Evolvability/Portability		Cost/Schedule	Negative	
Evolvability/Portability		Performance	Negative	
Evolvability/Portability		Usability	Negative	
Performance		Reliability	Negative	Kotonya and Sommerville (1997) These quality attributes are mentioned as non-functional requirements and are described in IEEE-std-830-1993
Performance		Security	Negative	
Flexibility/Runtime flexibility		Real time	Negative	Bosch 2000 pp 31 and 266 No reference to definitions for the quality attributes was given.
Flexibility/Runtime flexibility		Reliability	Negative	
Reusability		Performance	Negative	
Fault tolerance		Real-time computing	Negative	
Reliability		Correctness	Positive	McCall 1994, Encyclopedia of Software Engineering The definitions given by McCall are used.
Integrity		Efficiency	Negative	
Usability		Efficiency	Negative	
Usability		Integrity	Positive	
Maintainability		Efficiency	Negative	

Maintainability		Usability	Positive		
Testability		Correctness	Positive		
Testability		Reliability	Positive		
Testability		Efficiency	Negative		
Testability		Usability	Positive		
Testability		Maintainability	Positive		
Flexibility		Correctness	Positive		
Flexibility		Reliability	Negative		
Flexibility		Efficiency	Negative		
Flexibility		Integrity	Negative		
Flexibility		Usability	Positive		
Flexibility		Maintainability	Positive		
Flexibility		Testability	Positive		
Portability		Efficiency	Negative		
Portability		Maintainability	Positive		
Portability		Testability	Positive		
Reusability		Reliability	Negative		
Reusability		Efficiency	Negative		
Reusability		Integrity	Negative		
Reusability		Maintainability	Positive		
Reusability		Testability	Positive		
Reusability		Flexibility	Positive		
Reusability		Portability	Positive		
Interoperability		Efficiency	Negative		
Interoperability		Integrity	Negative		
Interoperability		Portability	Positive		

**Table 5: Stated relations between quality attributes.**

The influence between quality attributes, in table 5, is said to be positive or negative. It should also be possible to set the value to “no effect”, if the relation between two quality attributes are said to have no effect on each other. A perhaps risky interpretation is that relations not mentioned has no effect, it could also be the case that the relations have not been found or established yet. The literature read has though not dealt with that type of relation, except McCall who states that not mentioned pairs of quality attributes are unrelated.

A problem with this compilation of relations is that in some cases it is not possible to determine which definition of the quality attribute that is used.

Further the abstraction level for the relations is rarely stated, so it is hard to know the granularity of the relations as well. A reasonable assumption is that the relations described are generic relations, applicable for a majority of projects.

It is also worth noticing the fact that some of the authors are mentioning the relation casually, not explicitly declaring the relation. In some cases it is understood that there exists a relation and it is assumed that there is no further need to clarify it.

One way of dealing with the conflicts between quality attributes are to clarify the conflicts. Within the research material read for this thesis two methods were found for handling the conflicts. These will be described in the next section.

### 6.2.3 Handling the conflict

When handling a problem, for example conflicts between quality attributes and stakeholder interests, it is helpful, if not necessary, to have an extensive knowledge and information about facts. If there exists a conflict or opposing interests, the solution will in most cases lead to prioritization and a compromise. If this prioritization is done in a bad way without monitoring or having any understanding for potential consequences, problems will most likely arise.

The solution today, described in literature, is to build a framework and monitor the relation within your systems. The framework will help to build a model over the quality requirements or non-functional requirements. This model will be completed with partial quality goals and with ways to fulfill the partial goals. This method is further described in Section 6.3.

Another way to handle the problem is to create facts and knowledge. This approach is taken in the SQUID (Software Quality In Development project) approach, by aiming to define/refine product non-functional requirements in measurable terms such that a full quality requirements specification identifies quantifiable target values for each of the specified non-functional requirements (Kitchenham et al., 1997). Tom Gilb also takes the approach of making quality requirements measurable (Gilb, 1997). Gilb also has some principles, “The critical control principle: All critical attributes must be specified and controlled throughout the project and product lifetime.” and “The measurability principle: All attributes can and should be made measurable in practice”. A conclusion of these principles is that you must control your quality attributes, and by controlling means being able to measure the quality and effects. It has also been suggested that a number of values should be presented and monitored for quality attributes. These values are:

- Scale, which defines the measurement scale for the attribute.
- Test, how the value should be tested in practise.
- Worst limit, which sets the lower or worst limit accepted for the quality attribute. Below this limit the quality attribute will not be satisfactory.
- Plan, the planned level for achievement.
- Best limit, the optimal limit set for the quality attribute.

These values may also be accompanied by a current status, which is the achieved level at the moment. This can be used for comparison to worst and planned limits and also as input to the prioritisation process, if five out of seven quality attributes are between the worst and planned levels, and the other two are below worst limit, this would give a pointer on which two attributes to continue working on.

Over time one important lesson is learned: there is no magic that will solve all problems directly, and to be able to create a working solution you need to join different techniques, tailor them and tune them for your own organisation.

## **6.3 State of the art**

This section presents findings that are deemed representative for the academia viewpoint of what is the right thing to do based on the selection in the sample for the literature survey.

### **6.3.1 The conflict**

There are research made that supports the existence of relations between quality attributes. These relations cause conflicts between the attributes, which in turn represent conflicts or different priorities in stakeholder interests. It is not sound to discharge these conflicts and not monitor the consequences and by that looking the other way.

The way of handling the relations and potential conflicts is by increasing knowledge about the consequences. If one quality attribute is prioritized another, perhaps equally important attribute, may have to be omitted. What is likely to happen is that the focus will lie on the quality attributes that are interesting for the current phase, or stage in the life cycle and future quality requirements will be out of focus and cause problems further ahead.

This leads to the importance of relations between quality attributes that are currently handled as well as monitoring relations between present and future quality attributes. To be able to do so you need to find the relations specified for your organization and operation.

### 6.3.2 Finding relations

As described by Chung et al. (1999) building a framework and a knowledge base for quality attribute relations is preferable. This will support finding the relations as well as building knowledge about conflicts and consequences.

An introduction to the framework and its usage are given here. For more detailed information see Chung et al. (1999).

The framework uses a term called softgoal, softgoal represents a goal that has no clear-cut definition and/or criteria as to whether it is satisfied or not. The softgoals represent the non-functional requirements for the system.

When using the NFR (Non-Functional Requirements) Framework there is a number of steps that needs to be taken, the most important ones are the following:

- Acquiring or accessing knowledge about:
  - The particular domain and the system, which is being developed.
  - Functional requirements for the particular system.
  - Particular kinds of NFRs and associated development techniques.
- Identifying particular NFRs for the domain.
- Decomposing NFRs.
- Identifying “operationalizations” (possible design alternatives for meeting NFRs in the target system).
- Evaluating the impact of decisions.

These steps intend to gain knowledge, and decomposing the high level softgoals into more detailed quality goals. To be able to execute these steps, questions need to be answered. To aid the question process, catalogs are used. The NFR Framework uses these types of catalogs. One handles the particular type of NFRs, they are focused on for example performance or some other quality attribute. The second type deals with the development technique or development method, and answers the question which technique is suited for fulfilling which type of requirement. The third catalog type handles the implicit relations among softgoals; these implicit relations are also called correlations and trade-offs.

The softgoal handled may be ambiguous, which is no good, the NFRs way to solve this problem is to decompose the softgoals into smaller narrower parts. When the overall softgoals are decomposed into smaller unambiguous quality goals the next step is to make prioritization among these goals. Here the stakeholders and other representatives are consulted in order to make a balanced decision of which attribute or attributes that should be prioritized.

The next step is to choose among the operationalizations for achieving the goals. Also the operationalizations can be broken down into smaller parts as well. The influence of the operationalization and the quality goal is also represented, if there is a positive or negative influence. Next is to describe implicit interdependencies between softgoals.

The information needed to carry out these tasks is fetched from the catalogs described earlier. All of these are also documented in a graph, called Softgoal Interdependency Graphs, SIGs. There are also a notation defined and described to draw these SIGs and how the connections are to be notated. More information can be found in Chung et al. (1999).

In short, the NFR framework for handles quality goals and decomposing them, accompanying the softgoals with ways to solve them, monitoring the relations between the qualities and also the solution influences on the softgoals.

## 7 PRACTICE

This section is intended to describe the status of a typical software company today. Which quality attributes are used within their organization and how are the measuring and controlling carried out and which relations have been noticed?

The survey conducted in this thesis has been designed to find answers to this line of questioning. This is based on a presumption that all quality attributes described in literature, are not presently acknowledged within industry. Further the measurement process and method are not as rigorous as suggested from academia. This in collaboration will lead to that not the same relations are found within industry as academia are said to have found.

In a wider perspective the industry survey is also intended to find out what the practitioners are doing, what are their problems, and to see if the solutions presented in literature are applicable in industry.

The scope of this survey in terms of companies interviewed can be found in Section 5.2.2. The scope of handling business strategies and other economical interests was left out; the survey was only focusing on usage of quality attributes, measurements and potential relations and conflicts through the system lifecycle.

### 7.1 Introduction

An opinion is that industry is dealing with business and academia is dealing with problems. A coarse simplification is that you receive appreciation in industry by making good business, and in academia by solving a problem. The solution found by academia must not always be economically realistic.

When looking at industry business is everything, which places a different focus on their operation and every day life. There must exist an economical benefit, short term or long term, for their actions. This will create a new set of problems that perhaps are not always acknowledged within academia and described in literature. This is why the survey within industry is equally important as interesting.

### 7.2 Result

The result is presented in three sections:

- Quality attributes used in industry.
- Quality measurements related to the used quality attributes.
- Industrial opinion of relations among quality attributes.

A remark that is suitable to make is that some of the companies interviewed were mainly consulting companies and by that often working under the customer's directive. This will in some cases affect the answers depending on that they do not have any own collection of measures and processes when they are often obligated to use the customer's processes.

#### 7.2.1 Quality attributes used

As anticipated there are not always a clear understanding of the quality attributes and their meaning within industry, this is not connected to competence differences, but are more likely connected to the actual usage of the terminology. A way to avoid the problem was to introduce the quality attributes definitions stated by McCall (1994) to the interviewees when conducting the interview. There was naturally an option to state and describe a different quality attribute if the set given by McCall were not suited.

The result showed that the two most popular quality attributes was Correctness and Efficiency. Six out of eight interviewees mentioned these quality attributes. The three least used quality attributes are: Integrity, Testability, and Reusability.

In table 6, an X means that the interviewee responded that the quality attribute was used.

Interviewee	1	2	3	4	5	6	7	8
Correctness	X	X	X		X	X	X	
Reliability	X		X		X	X		X
Efficiency	X	X	X		X		X	X
Usability	X	X	X			X		
Integrity	X		X					
Maintainability			X			X		X
Flexibility	X		X			X		
Testability			X			X		
Portability	X	X	X			X		X
Reusability	X					X		
Iteroperability	X		X			X		

**Table 6: Quality attributes used in industry.**

The persons interviewed also missed some quality attributes, the most frequently mentioned is time to market or project precision, which means two things, on how short time is it possible to develop the system and with which precision is it possible to deliver. This was found necessary in order to be able to interact with other projects, which are delivering other parts of a common larger system.

It should be noted that some of the answers were vague, in the sense that they are using the quality attribute, but it is not explicitly controlled or measured and the overall focus is not that good. But the question in the design was formulated to allow for this type of answer with the objective to grasp as many quality attributes as possible.

### 7.2.2 Measurement of quality attributes

As anticipated not all quality attributes were measured, the two most used quality attributes were also the measured ones, i.e. Correctness and Efficiency.

Correctness was typically measured by counting the errors found during testing and sometimes during operation. In order to get the measure fault density, the number of faults was usually divided with lines of code or some other size measure for the source code.

Efficiency was the other frequently measured quality attribute. Two aspects of efficiency were measured: throughput and response time. Throughput was mainly focused on how many transactions, of some sort, could be handled per time unit, the handling could mean for example logging or computing and generating output. The response time was generally measured with the number of connections, requests about an answer onto the system per time unit.

Other measured quality attributes were:

- Project precision, measure on delivery accuracy.
- Time to customer, the time needed to deliver the product.

The majority of the measures were conducted as quantitative measurements, by counting something and putting numbers on the result.

A common observation is that measuring seems to be problematic and time consuming. The majority of the interviewees were expressing the difficulty with executing measurement and gathering metrics. The difficulty of measuring was the main reason to why they are not measuring to a greater extent. But there were also innovative solutions, examples have been given that the quality attributes were transferred to functional requirements, or that they were measured through another phenomena in the process. Two examples were given for maintainability. One way of handling maintainability was to transfer the fulfilment of maintainability to the existence of design, and source code documentation, and by this making maintainability controllable and measurable. Another way was to measure the turnaround time to the customer; if an error was reported from the customer the time was measured on how long it would take before the error was corrected and the system

was updated at the customer's site. A long turnaround time may indicate low maintainability of the system. This is said to be an indirect measure, and further the term long was not defined. A guess is that turnaround time is estimated and becomes long when the estimations are exceeded.

These ways of measuring maintainability are not optimal, mostly because maintainability is measured after the system is built, when there is little possibility to corrections in the system. But never the less, they are providing a way to control and measure maintainability, which is beneficial.

It should also be noted that there existed a spectrum of how accurate and rigorous the measures were collected, for some quality attributes there existed a metrics process and for others there was simply an indirect measure or a quick evaluation as the measure.

The measures have not a great value of their own, the value of measurements comes when they are used as input to improvement processes or as input to coming projects. The usage of the measures was also varying between the companies interviewed, some were using the measures for monitoring the development process, if a quality dropped dramatically, something was wrong and the error was sought. Others were simply stashing the measurements with no particular use for them. It is also necessary to measure the same way throughout the project and in different projects so that the measures can be compared with each other. This also varied according to the interviewees, it was not sure that there are consistent measures available.

Measurements are a basis for making improvements, monitoring the process and judging the changes or improvements made to the process if they were beneficial or not for the desired quality attribute. The measurements will also provide information to future prioritisations between quality attributes. It is necessary to have metrics on how a prioritisation of one quality attribute will affect others. In order to make the correct decisions measures need to be collected so that relations can be established and not solely based on beliefs.

### 7.2.3 Relations found in industry

One hypothesis was that there exist relations between quality attributes, this hypothesis was both confirmed and denied by industry. When asking bluntly if they believed that there were any relations between quality attributes, a majority answered that there were. While some insisted that the relations do not depend on the quality attributes, but instead on how they are realized in the code and then especially related to code size.

Both ways of addressing relations between quality attributes can be correct. When looking on a higher abstraction level the possibility is that quality attributes relate to each other, depending on the common realization of the quality attribute. And when seeking more in detail perhaps the code size is more the common glue that creates the relation between quality attributes. It is also possible that other solutions may influence the relations, such as design methods, how the memory is allocated and so forth. Though these objections could be raised, relations among quality attributes were agreed upon and the relations stated.

It was not an easy task to extract some explicit relations from the interviewees; this was based on that there was no explicit knowledge about the relations; the answers were therefore based on experience from the interviewees. It is worth noticing that industry as well as academia seems to have a focus on the negative relations, the negative relations were dominating. But there was also an interest of knowing the positive relations, so the possibility to achieve higher level of quality for a number of quality attributes to low extra cost and effort, depending on that the ground was already in place due to a required quality attribute. The most common relations found within industry are the relation between the following, Correctness and Efficiency, which is negative, and Usability and Efficiency, which is negative, and Usability and Reliability which is positive, and Time to Market and Correctness, which is negative.

The complete set of relations found is stated in the table below, the quality attributes are defined after McCall's definitions. The term Time to Market (TTM) is also used. This is one of the quality attributes or limitations given to the system within industry. TTM states the time (lead time, calendar time) needed to deliver the system to the market or customer. The quality attributes definition from research has not mentioned this attribute or limitation.

Interviewee	QA	vs.	QA	Relation/Influence
1	Flexibility		Efficiency	Negative
	Testability		Correctness	Positive
	Usability		Reliability	Positive
2	Time to Market		Correctness	Negative
	Time to Market		General quality <sup>*</sup>	Negative
3 <sup>**</sup>	Usability		Reliability	Positive
	Reliability		Maintainability	Negative
	Reliability		Correctness	Negative
	Usability		Efficiency	Negative
5	Time to Market		Maintainability	Negative
	Reliability		Maintainability	Negative
	Reliability		Efficiency	Negative
6	Correctness		Efficiency	Not stated <sup>***</sup>
	Efficiency		Maintainability	Not stated
	Maintainability		Testability	Not stated
	Portability		Maintainability	Not stated
	Interoperability		Efficiency	Not stated
7	Correctness		Efficiency	Negative
	Usability		Efficiency	Negative
	Time to Market		Correctness	Negative
	Usability		Correctness	Negative
	Usability		Time to Market	Negative
8 <sup>**</sup>	Portability		Maintainability	Negative
	Availability		Maintainability	Negative
	Portability		Correctness	Negative
	Availability		Correctness	Negative
	Portability		Efficiency	Negative
	Maintainability		Correctness	Positive
	Availability		Efficiency	Negative

**Table 7: Relations found within industry.**

What this table shows is that there is a number of relations and the different organizations are handling their own set of relating quality attributes.

What was agreed upon within industry was the fact that the relations were dealt with in the prioritization of which quality attributes should be maximized or improved and which should not. But whether the relations were handled in the requirements process the opinion varied, about in half of the organizations, according to the interviewed persons, handled or acknowledged the relations in the requirements process. The opinion that relations should be omitted from the requirements process was also mentioned, with the motivation that, all requirements should be brought

<sup>\*</sup> This person had a distinct opinion that time limitations had a negative impact on overall quality.

<sup>\*\*</sup> The opinion of this person was that the relations are based on the code size, and are not solely depending on the quality attributes.

<sup>\*\*\*</sup> The statement "Not stated" depends on the reluctance to explicitly give the relations influence, the relations were though acknowledged.



forward without bothering with potential limitations. The opposite was that any decision about prioritization and ranking of non-functional requirements should be made as early as possible, and also avoiding asking impossible requirements.

A common apprehension is that the relations cause problems, and that knowledge would help to make informed decisions and avoiding consequences later in the system's life cycle. A reflection is that the industry, as well as academia, is focusing on the negative relations instead of making use of existing relations and increasing several quality attributes at a low cost. For example, high priority for a majority of the companies was the attribute correctness. The relation between maintainability and correctness was said to be a positive relation, as well as the relation between testability and correctness. This would make it possible to improve maintainability to a low cost when achieving correctness.

The industrial opinion is also that they know which quality attributes that are important for their operation and their customers. The main issue is how to control the quality attributes, i.e. how to quantify and measure quality.

### **7.3 State of the practice**

The current state for the practitioners is that relations are acknowledged, there are varying opinions on how the quality attributes relate, through generic relations among the quality attributes or through their realization in the form of source code size for example.

The relations are causing conflicts and also causing consequences further down the line that are hard to comprehend at the moment for the decision. There is a united opinion that extended knowledge covering both generic and specific relations are needed. Either for usage at projects level or at management level in order to clarify relations and potential consequences.

The additional knowledge would primarily be used for decision making, but also for avoiding conflicts and increasing adjacent quality attributes to low cost and effort.

It is also clear that industry is faced with other restrictions not always visible in academia, such as time to market demands and more specific requirements. The developing industry or organization is not always given the opportunity to make the best solution possible, based on decisions taken on higher management level. Some quality attributes are also set in stone, which means that these may not be altered, decreased or negotiated in any way. Requirements can some time not even be negotiable, no matter what the consequences or time needed for implementing a suitable solution. Examples of such requirements are laws and regulations that the customers need to follow and then transfer down to the system.

The main problem with gathering more information about the relations from an industrial perspective is the possibility to control and measure quality. There have been some innovative steps taken by companies to solve this problem as well, simply by taking the non-functional requirements and transforming it into functional or process requirements. An example is: the quality attribute maintainability is fulfilled by the existence of design and code documentation, the quality attribute integrity is satisfied by the existence of specified functions and functionality. This was an example of a solution to the measurement problem.

A problem that exists within industry is the motive to find solutions on how to actually use knowledge about relations and conflicts, how to gather the information needed to improve the processes and achieve higher quality. There is in no way lacking knowledge or competence for doing this. A guess is that time is the limiting factor and that the focus is on delivering products with enough quality, and not maximized quality. This depends probably more on business strategy and time limitations than anything else, and business strategy and marketing decision are not the scope of this report, but it is not unlikely that it has a serious impact on software quality.

## 8 ART VS. PRACTICE

As mentioned in previous sections there is a number of problems but also solutions, it is not always that academia and industry agrees upon the same problem or the solution. But there is common ground as well.

One point that industry and academia agrees upon is the existence of relations among quality attributes. The way to address the problem varies; academia has a rigorous approach to mapping the relations and also used clear-cut definitions, while industry is more working with non-specific knowledge and is to a greater extent trusting the individual developer or project manager to make the right decisions. In short industry is more experience based than academia.

What is needed from an academic perspective is to provide industry with tools and information on how to extend their knowledge and to make their knowledge explicit and move away from gut feeling and some times guessing. Academia can also take a further look into the actual relationships, what is the construction that makes quality attributes relate to each other. A step in the right direction is the NFR Framework described in Chung et al. (1999), which offers a method for creating explicit knowledge based on the own organization.

What industry needs to do is to prioritize the quality aspects and further acknowledge the relations between quality attributes. This has to be done on all levels of the organization, from developer to manager. There is a necessity for looking and assess the consequences of the decisions taken to a greater extent and also monitoring the potential changes and consequences over a longer time period. The quality demands are not likely to drop or even stagnate, the likely direction is that the customers will become more interested and demanding as well as that the surroundings of the system will change more rapidly.

Industry and academia are in symbiosis with each other, to do good research academia needs industry for case studies and also for providing interesting problems and challenges. What industry needs to is to clarify the problems and perhaps elevate the abstraction level so that academia can attack the problems. By this better research can be done in the areas as well, and perhaps more tailored research applicable to problems present in industry. This will hopefully lead to improved quality, both for research and for the delivered software systems.

## 9 CONCLUSIONS

Industry has a desire to know more about the relations among quality attributes, what is missing is how to gain the information from academia, which is not always explicit and concrete.

It is found that relations between quality attributes create conflicts within the organization as well; it is not always possible to make a decision that suits all stakeholders of a system. The problem of relations and prioritization are handled in the prioritization phase between requirements, both functional and non-functional. There is a desire as well for handling the relations as early as the requirement process, in order to reach a decision as early as possible.

It is not likely that quality demands will decrease, and in order to sustain and increase quality you need to have knowledge about the relations, if not you may place yourself in an impossible situation where quality requirements are in opposition to each other and you are not able to increase any of these, or worse the relations between quality attributes will lead to descending quality. It is necessary for the practitioners to gain knowledge, both from their own organization and from academia to avoid the pitfalls. To gain knowledge about the relations from organizations, it is necessary to start monitoring what happens with quality based on the decisions taken, what are the effects?

When the information is in place it can be used to inform higher management, and other parts of the organization and indicate the potential consequences. With this information it is possible, for all parts of the organization, to make an informed decision and by that avoiding the consequences and achieving the quality goals.

As anticipated and stated in the hypotheses, the set of quality attributes was not the same for industry as for academia, the set within industry was neither as extensive and nor so detailed and measured as described in academia. Though, the industrial opinion is that the quality attributes they are using generally satisfies their needs. A common opinion was that the attributes used were enough, but could be improved in the way that they were controlled and measured. The level of achievement for these attributes was judged as unsatisfying by industry themselves.

The hypotheses were in general found to be true,

- Industry only uses a subset of the set of quality attributes given by academia.
- All quality attributes are not measured within industry.
- The attributes used in the software engineering process are controlled in some way, not necessarily measured. This is though not always the case, all used quality attributes were not measured or controlled.
- The existence of quality attributes within industry was roughly the same as for the academia. Though some qualities that describe time to market and project precision do not commonly exist in the research material covered in this thesis.
- Not all relations are acknowledged in the requirements phase for a system.
- And finally there are relations among quality attributes, and they cause conflicts within the software engineering process. Though some objections towards the causality of these relations were stated, and the theory that the relations depend on the realization of the quality attributes instead.

By this it is safe to draw the conclusion that further work needs to be done, and that there are conflicts and trade-offs made today depending on relations between quality attributes.

It is also noted that the positive influences between quality attributes is not investigated and used in the same extent as the negative influences. This is unfortunate, by exploring the positive relations, it would be possible to extend the total

quality of the software with only a small contribution of effort, the exchange for this moderate investment could be rewarding. But more work, research and cooperation are needed to cover this area.

This thesis work has taken the approach to find and confirm relations by using surveys. An alternative approach would be to take a deeper look at the realizations of the quality attributes. And by looking at the differences of the realization, a more detailed and perhaps accurate picture of the relations could be drawn.

A question is to know on which granularity level it is interesting to know about the relations. And in which phases of the process is this information actually needed, perhaps it would be suitable to have different levels of accuracy of the relations for different phases in the software engineering process.

The probably most suitable solution today is that the organization itself is trying to build this knowledge, as they are the experts on knowing when and what information is needed, this approach is taken by the NFR Framework, described in Section 6.3.2, and it could be a good starting point.

There are still issues to work with concerning quality attributes, conflicts and trade-offs. What is needed is continuous work from academia to supply industry with research and also facilitating the usage of research. A starting point would be to further investigate and clarify the relations between quality attributes. What is the substance of the relation, is it the code size or design solution, or something else.

Industry would benefit from gathering knowledge about relations present in the systems constructed within the organization. A step towards this is to start controlling the quality, by measuring and actually using the measures as indicators and input to coming project and improvement suggestions. When measures are in place monitoring of consequences and the effects of relations is possible to do, this may lead to that some new relations are found and others are discharged.

If efforts from both industry and academia are combined more fact about the relations between quality attributes and the conflicts thereof are established.

The intention with this thesis work was that some questions were to be answered and some issues, if not solved, at least clarified. And hopefully the reader agrees after reading this master thesis work.

## 10 REFERENCES

- Boehm B. In H. 1996, 'Identifying Quality Requirement Conflicts' *IEEE Software* pp., 25 – 35, March 1996.
- Boehm, B. 1988, 'A Spiral Model of Software Development and Enhancement' *IEEE Computer*, pp 61 – 72, May 1988.
- Bosch J. 2000, *Design & Use of Software Architecture - Adopting and Evolving a Product-line Approach*, Addison-Wesley, Great Britain.
- Bøegh J., Depanfilis S., Kitchenham B., Pasquini A. 1999, 'A Method for Software Quality Planning, Control, and Evaluation' *IEEE Software* pp., 69-77, March/April 1999
- Chung L., Nixon B.A., Yu E., Mylopoulos J. 1999 *Non-Functional Requirements in Software Engineering*, Kluwer Academic Publishers, Boston, USA.
- Ericsson 2001, <[www.ericsson.se/softwaretechnology/](http://www.ericsson.se/softwaretechnology/)> (2001-06-06).
- Europolitan 2001, <[www.europolitan.se](http://www.europolitan.se)> (2001-07-27)
- Gilb T. 1997, *Principles of Software Engineering Management*, Addison-Wesley Longman Limited, Harlow, England. 1997
- Häggander D., Bengtsson PO., Bosch J., Lundberg L. 1999 "3rd International IASTED Conference on Software Engineering and Application" Scottsdale, USA 1999. Published in "Software Design Conflicts – Maintainability versus Performance and Availability" Department of Software Engineering and Computer Science, Blekinge Institute of Technology Dissertation Series No. 02/01.
- Kitchenham B., Linkman S., Pasquini A., and Nanni V. 1997, 'The SQUID approach to defining a quality model', *Software Quality Journal*, pp., 211 – 233, Number 6, 1997.
- Kotonya G. and Sommerville 1997, *Requirements Engineering*, Wiley, West Sussex, England.
- Maurer S.B. and Ralston A. 1991, *Discrete Algorithmic Mathematics*, Addison-Wesley, Massachusetts, USA.
- McCall, James A. 1994, *Encyclopedia of Software Engineering*, Volume. 2. John Wiley & Sons, USA.
- Micronet 2001, <[www.micronet.se](http://www.micronet.se)> (visited 2001-07-27).
- Pfleeger S. L. 2001, *Software Engineering - Theory and Practice, Second Edition*, Prentice-Hall, USA.
- Robson C. 1998, *Real world research*, Blackwell Publishers Ltd. Oxford UK.
- Symbian 2001, <[www.symbian.com](http://www.symbian.com)> (2001-06-06).
- Teleca 2001, <[www.teleca.se](http://www.teleca.se)>(2001-07-27).

ISO/IEC 2000 “Information Technology – Software Product Quality – Part 1: Quality Model” ISO/IEC FDIS 9126-1:2000(E).

## **Appendix**

Appendix A is the survey design, enclosed after this page.

# Survey design

Kennet Henningsson, BTH, IPD

## 1. Survey design

This document describes the intention, methods and sample for my intended survey.

## 2. Research question

The overall research question is “Which are the characteristics for the set of quality attributes, which quality attributes does it contains and what are their relations?”

Requirements for a quality attribute to be part of the quality vector is that the quality attribute or quality factor is used in some way in the development/requirement/evaluation process. The term *used* will be further explained later.

### 2.1 Survey aims

This survey aims to cover the main part of the research question, from the industry perspective. The usage, the relations and the measurements.

### 2.2 Survey issues

The survey has three issues to cover

- Quality attributes usage in industry, which are used.
  - Are the term quality attributes familiar to the organization?
  - Which quality attributes are present in the software engineering process?
- Industries opinions of relations among quality attributes.
  - Do you sense any relations between the quality attributes?
  - How would you characterize the relation, weak, medium, or strong?
  - Is the relation stringent or working in opposite direction?
- Quality measures related to the used quality attributes in industry.
  - Is the quality attributes mentioned controlled in some way?
  - How is the control done?
  - Is the result of the control qualitative or quantitative?

These issues are supported by definitions of quality attributes or quality factors.

### 2.3 Expected results

Expected results is a consensus picture, from industry, of which quality attributes that are part of the engineering and development process and thereby seemed as useful. And also give a gathered view of their relations.

The quality attributes are useful for the organization in their process of describing quality of their developed systems from different angles. For example from the users perspective with quality attributes such as usability and stability.

The answers from the interviewees are supposed to contain the usage a number of attributes. The anticipated quality attributes are described in table 1 along with a short motivation to why they are anticipated to be present. The quality attributes are based on McCall’s definition of quality attributes (McCall, 1994).

Quality attribute	Motivation
Efficiency (Performance)	It is common that the quality of a software system are judged by the speed for which it completes the task.
Maintainability	At some point in time the system will be maintained in some way,



	and the ease of the maintenance are probably of interest.
Reusability	It is of interest for companies to reuse former developed code as well as it is of interest to develop components or system parts that can be reused.
Integrity (Security)	Today the problem with intrusion are well known and are probably a ground for judging the quality of a software system.
Time To Market (This quality attribute are not present in McCall's definition, (McCall, 1994))	There is a high probability that the industry will assess their systems quality from the grounds of how soon the system can reach the market. How quick will the solution be to implement.

TABLE 1. Anticipated quality attributes

## 3 Assumptions and hypotheses

### 3.1 Assumptions

There are some assumptions made, these are listed here.

- The people that will take part in the interview will be familiar with the term quality attribute or quality factor.
- The organization participating have normally handles the terms quality attributes or quality factors or some other term with similar meaning.
- The organization participating have some way of controlling, if not all, at least some of the quality attributes.
- The industry have created some of their own quality attributes based on the industrial reality.

### 3.2 Hypotheses

The following hypothesis are the basis for the line of questioning for this survey.

- The industry only uses a subset of the quality attributes described in literature. Not all quality attributes are taken into consideration with in the process
- All the quality attributes are not measured. Some are measured quantitatively, some are measured qualitatively and some are not measured at all.
- The attributes that are used in the software engineering process are also controlled in some way, not necessarily through quantitative measurement.
- There exists quality attributes within industry that are not described in literature, depending on the marketing demands set on the industry. These attributes are not necessarily mentioned as quality attributes, but they are factors that influences the development of software.

## 4. Strategy and method

The chosen strategy is survey. The motive is that survey are collecting information from a group of people in a standardized way, where you control the sample and which questions to ask. A survey can be both qualitative and quantitative, depending on the method used for the survey. The survey approach can also cover a great variety of questions, such as 'how', 'what' and so on.

The chosen methodology will be semi-structured interviews. These are found to be useful when extracting a qualitative information from an individual. Also the line of questioning that I assume to be using will not be suitable for a pre-determined answer schedule. I anticipate that the exact wording and meaning of words will not be the same for all the interviewees, which is a prerequisite for using a standardized schedule for the answers. (Robson 1995)

### 4.1 Threats

The following threats are identified for this particular survey.

- Time limitations, there is a risk that the interviewee has a narrow schedule and by that do not have

the time necessary to carry through a meaningful interview.

The threat will be addressed by making a dry run interview to ensure that the interviews can be conducted within the agreed time. Further interruptions are avoided by locating the interviews in a conference room without phone and mail, and asking the interviewee to turn off mobile phones.

- The results from the survey are influenced greatly by the respondent, his opinion, knowledge, experience, personal agenda concerning the subject and interest of taking part in the study. This survey aims to gather the interviewees current understanding of quality factor relations. And hence the opinions, knowledge and experience does not pose a threat, but is rather part of the results. Personal agendas and interest of participating can be disclosed by asking questions about the subject relevance, e.g. “do you think it is meaningful to investigate relations between quality attributes?”.
- The result are compromised by the interviewer, by his bias, his characteristics and perhaps also interpretations of the given answers. These threats are avoided by peer reviewing the questions, answers and the conclusions of the survey during the interview or in retrospect. In addition the interviewees are asked to acknowledge their recorded answers.
- Depending on the sample, the results generalizability may be threatened, i.e. the findings may not apply to the general case. This threat are addressed by choosing a sample that contains of companies with varying types of development projects e.g. information systems, web-systems, real time systems and so on. Further to avoid a single point of influence are to have more than one person from each organization in the sample.

## 5. Sample

The sample will mainly be conducted as convenience sampling depending on that the sample will consist of companies that BTH (Blekinge Institute of Technology) already have co-operation with. Which is convenient in this case, the companies are not necessarily located adjacently but the communication routs are established.

### 5.1 Companies interviewed

The sample consists of the following companies.

- Ericsson Software Technology (Ericsson, 2001). Ericsson develops operational system for telecommunications and mobile systems.
- Symbian (Symbian, 2001). Symbian develops operating system and applications for wireless and handheld units primarily.
- Volvo IT, (Volvo IT, 2001). Volvo IT is mainly a support organization to Volvo and develops administrative support system for the production organization.
- Micronet (Micronet, 2001). Micronet develops applications and products for the internet and mobile community, both by partners and in house.
- Europolitan (Europolitan, 2001). Europolitan develops administrative systems to the own organization as well as mobile services for the end customer using mobile devices within their mobile net.
- Teleca (Teleca, 2001). Teleca develops systems and services for most of the areas within software engineering. Both wireless, as integrated systems and drivers and hardware near solutions.

### 5.2 Who to interview?

Since my interest are in quality attributes and their possible relations, the sample will consists of persons in roles in the company that are likely to have knowledge about these possible relations.

Some suggestions for roles:

- Chief software architect, software architect.
- Quality assurance and quality management personnel.
- Project manager, with technical interest.
- Project manager, with economical interest.

The motivation for this sample is to receive influences from different parts of a project organization. The software architects have knowledge about many projects and also great knowledge about the quality

attributes that are interesting for this survey. The quality personnel has insight in the total quality of the software and how the quality are measured and monitored. The project managers contributes with knowledge about decisions made during the development project, also they may be aware of the trade-off made in order to comply some other prioritized quality attribute.

The selection of persons to interview will be done in consultation with interviewer and interviewee, it is favorable if the interviewee feels, after a short explanation of the research, that he/she are motivated to participate and has some input to the survey.

### **5.3 Prerequisites**

The company as well as the person that hold the position, as mentions in section 3.1, must fulfill some prerequisites in order to make the interview meaningful. This might be some what unorthodox to make such demands, but it is necessary.

#### **5.3.1 Company prerequisites**

- Are familiar with the concept of quality attributes, quality factors or similar terminology
- Have the positions/roles mentioned in section 3.1 or personnel interested in participating with motivation and input to the research area.
- Experience from software projects. The experience can be gained both as a developing organization and as a receiving organization.

#### **5.3.2 Personnel prerequisites**

- Taking part in software development project acting in their role.
- Basic knowledge about software quality, which is the target area for this survey.
- Desirable are also knowledge about the terminology quality attribute or quality factors.

## **6. Questions to be asked**

The tactic described in Real Work Research by Robson will be used. There Robson divides the interview in different sequences; introduction; warm-up; main body of the interview; cool-of and closure. This will influence the questions asked.

### **6.1 Coding of the questions**

The answer to the questions will be related to the quality attribute definition given by James McCall contained in Encyclopedia of Software Engineering (McCall, 1994).

### **6.2 Question areas and questions**

The survey are intended to touch the following areas, personal experience, quality measures, quality attributes used, quality prediction, quality evaluation, relation between quality attributes, how the relations has showed, measures taken to minimize the effects of the relation and personal reflections on the relations.

#### **6.2.1 Personal experience**

The line of questioning in this section are intended to characterize the interviewed person and his background.

The anticipated answers will be written in between << and >>.

- Which education has the interviewee?  
<< Type of education, name of the program, number of years. >>
- Prior working experience.  
<< The emphasis of prior employments. >>
- Experience in the present role  
<< For which time the interviewee has held the current employment, the the current working assignment. >>

## 6.2.2 Quality attributes participating in the software engineering process

The line of questioning from here on are the basis for the result of the study.

- Which quality attributes is part of the development process?  
By “part of” means, part of a process description, part of the de-facto usage, or are present in the mental awareness of persons involved.  
<< The names or a description of the different quality attributes or quality factors that are taking part of the software engineering process in the interviewees organization according to his interpretation.>>
- What does each of these quality attributes mean, which criteria does it contain?  
<< The answer would be the meaning of the quality attribute which make it possible to relate the quality attribute given to one present in the McCall definition, if not it will be labeled as a new unique quality attribute. >>
- Are there lacking any quality attributes.  
<< The answer will reveal if the interviewee are missing any quality attribute, which will imply that some factors in the software engineering process can not be described with the present quality attributes. >>
- Some examples of quality attributes that are not used.  
<< Will give some quality attributes that seems superfluous. >>
- Why are some quality attributes not used.  
<< The motive to why all quality attributes are not used, not needed, not applicable, too hard to use and so on. >>
- Are the quality attributes used satisfactory  
<< Will answer if the quality attributes used are enough or if even they do not match the needs within industry. >>

## 6.2.3 Quality measures

- Are the quality attributes measured or controlled in any way?  
<< Answer if there is any monitoring on the quality attributes in any way. >>
- Are the measures qualitative or quantitative?  
<< Indicates the type of control over the quality attributes. >>
- Is the measure or control essentially the same through out the majority of the projects?  
<< Will indicate if there is a possibility to compare quality attributes between projects. >>
- How are the measures used?  
<< Indicates how the metrics result are used, if it is integrated into the process or just a momentary indication on system quality. This will also indicate when in the process the metrics are used. >>

## 6.2.4 Relation between quality attributes

- Do you feel that there exists any relations or influence between different quality attributes?  
<< The interviewee gives his opinion of existence of relations between quality attributes. >>
- Have you noticed any relation or influence between different quality attributes?  
<< Give any personal experience regarding relations between quality attributes. >>
- Between which quality attributes have you noticed the relation?  
<< An elaboration of the answer to the prior question. >>
- Are the relations depending on which type of system that is currently built?  
<< The interviewee's opinion on if the relation are system dependent or not. >>
- Do there exist any examples?  
<< If the interviewee has knowledge of any examples. >>
- Does the relations cause any problems with the process?  
<< If the interviewee experiences any problems that originate from relations between quality attributes. >>
- Is it desirable to have/extend the knowledge about relations' quality attributes?  
<< If there is any desire from industry for a map out the relations. >>
- Are the relations observed in the requirements process?  
<< If the relations are taken into consideration in the requirement process, not signing on to impossible requirements. >>

- Are the relations observed in the prioritization process between quality attributes?  
<< If the relations are thought of further in the process. >>
- Would you look at the relations as a limitation or as assets?  
<< The view onto the relations, problems or not. >>
- Are the relations or influence documented in any way within the organization?  
<< If the organization has such a clear view of the relation so that they can be documented.  
>>
- If there is a relation between QA1 and QA2, could you say if that relation were to be positive or negative? Meaning if QA1 are increased and QA2 also increases, the relation was to be positive, and vice versa.  
<< Indicates if the interviewee has a sense for if the relations are stringent or not. >>

### 6.2.5 Personal reflections on the relations.

- What is your personal reflection of relations among quality attributes?
- Would it be suitable to have the knowledge about the common relations?
- Any other comments on the line of questioning regarding quality attributes?
- Anything you wish to add?

## 7. Requirements and tools

A naturally requirement will be available time with the interviewees, planned time consumption are 60 to 90 minutes, with the present number of questions.

For conducting this survey I will also need information or access to information about the personnel and their roles at the companies.

Tools for saving the answers would be a tape recorder, so that the discussion can keep momentum and not have to wait for the interviewer to write down the answers and comments. This naturally implies that the interviewee, and his company, accepts that the interview are recorded.

## 8. References

Ericsson 2001, [www.ericsson.se/softwaretechnology/](http://www.ericsson.se/softwaretechnology/) (visited 2001-06-06).

Europolitan 2001, [www.europolitan.se](http://www.europolitan.se) (visited 2001-07-27)

McCall 1994, *“Encyclopedia of Software Engineering”*, editor MacCormic Volume. 2. John Wiley & Sons, USA 1994.

Micronet 2001, [www.micronet.se](http://www.micronet.se) (visited 2001-07-27).

Robson 1995, Colin Robson, *“Real World Research”*, 1995 Oxford. 510p.

Symbian 2001, [www.symbian.com](http://www.symbian.com) (visited 2001-06-06).

Teleca 2001, [www.teleca.se](http://www.teleca.se) (visited 2001-07-27).

Volvo IT, 2001, [www.volvo.com](http://www.volvo.com) alternative “<http://www1.volvo.com/frameset.html?nav-Type=1&url=http://www1.volvo.com/home/portal.html>” (visited 2001-07-27).