Consolidating Different Views of Quality Attribute Relationships

Mikael Svahnberg, Kennet Henningsson
School of Engineering
Blekinge Institute of Technology
PO Box 520, S-372 25 Ronneby SWEDEN
{Mikael.Svahnberg | Kennet.Henningsson}@bth.se

Abstract

In recent years, quality attributes have received increased attention as being critical for a software system's success or failure. Different classifications of quality attributes frequently mention many relations both positive and negative between quality attributes. However, the classifications and the quality attribute relations are often presented in such a way that they are not easily compared with each other, which means that it is difficult to confirm the relations found in one source with the relations found in another. In this article we triangulate between different sources where quality attribute relations have been (or can be) expressed. The contribution is a consolidated view of which quality attributes that are considered (by several sources) to have positive or negative relations with each other.

1 Introduction

When developing software it is vital to achieve a sufficient level of quality and precision of the product and process, i.e. that the developed system achieves its quality goals and that the quality goals are achieved within the given time frame and budget.

To accomplish this it is necessary to understand what factors affect the quality of the software. Previous studies (e.g. [13]) have shown that one cannot expect developers to share a similar view of the benefits and liabilities of software architectures. Factors such as the organisational role of a developer and previous experiences play an important role when assessing a software architecture based on quality attributes. From this study and others (e.g. [8, 6]), we conclude that the views expressed by e.g. McCall [10] about perceived relations between quality attributes may not be as commonly known or accepted as is generally held to be true. Hence, there is a need to further investigate different quality attributes and how these relate to each other.

In this article, we triangulate between different views, created for different purposes, of relations between quality attributes. The contribution of this is that it creates an increased understanding that may be used when deciding which quality attributes to prioritise in the development of a software system.

The remainder of this article is organised as follows. In Section 1.1 we further describe the background that has led up to this article, and present the scope of this study. We also present quality attribute relations found in a previous industry survey. In Section 2 we extract quality attribute relations from an architecture assessment study conducted in academia. Section 3 compares the quality attribute relations found in several literature sources, translating them into a single classification standard (i.e. [7]). In Section 4 we compare the three views (i.e. literature[1], industry and academic), looking for cases where the different sources confirm or confute each other. Finally, the article is concluded in Section 5.

1.1 Background and Scope

In previous studies the authors have investigated various approaches to identifying relations between quality attributes in general, applied in different architectures, or relations found in literature. These different studies relate to each other as illustrated in Figure 1. In this figure we see the different viewpoints from previous studies in industry [6] and academia [14] and from literature [5] (marked with 1, 2 and 3, respectively in the figure).

Study 1 [6] is a survey where eight employees at five industry partners rated relations between quality attributes (using McCall's [10] classification). From this study we intend to use the presented set of relations identified by at least two of the interviewed persons. These relations are presented in Table 1.

[1] It should be noted that the views in literature sources may also have been based on industry or academic studies.

978-1-4244-3723-8/09/$25.00

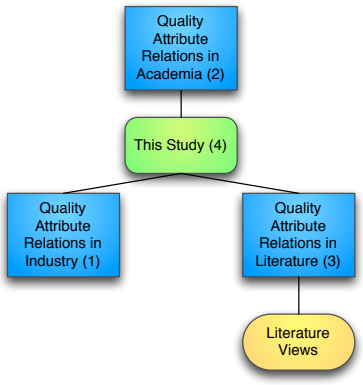

Figure 1. Previous Studies and their Relation

Table 1. Relations from [5]

Quality Attribute	vs Quality Attribute	Relation/Influence
Usability	Reliability	positive
Time to Market	Correctness	negative
Reliability	Maintainability	negative
Usability	Efficiency	negative
Correctness	Efficiency	negative
Portability	Maintainability	negative

Table 2. Ranking of Software Architectures per Quality Attributes (FQA)

	Microkernel	Blackboard	Layered	Model-View-Controller	Pipes and Filters	Sum
Efficiency	**0.274**	**0.186**	**0.110**	**0.121**	**0.309**	**1**
Functionality	0.228	0.261	0.179	0.188	0.144	1
Usability	**0.121**	**0.134**	**0.274**	**0.319**	**0.152**	**1**
Reliability	0.207	0.103	0.270	0.239	0.180	1
Maint.	**0.124**	**0.171**	**0.283**	**0.198**	**0.225**	**1**
Portability	0.194	0.0767	0.366	0.157	0.207	1

Table 3. Ranking of Quality Attributes per Software Architecture (FAS)

	Microkernel	Blackboard	Layered	Model-View-Controller	Pipes and Filters
Efficiency	**0.204**	0.170	**0.0714**	0.0564	**0.211**
Functionality	**0.138**	0.284	**0.109**	0.199	**0.152**
Usability	**0.0963**	0.126	**0.102**	0.222	**0.0980**
Reliability	**0.137**	0.0711	**0.106**	0.132	**0.132**
Maint.	**0.182**	0.254	**0.278**	0.230	**0.251**
Portability	**0.244**	0.0953	**0.333**	0.160	**0.156**
Sum	**1**	1	**1**	1	**1**

Study 2 [14] is a study where a number of architecture patterns (from Buschmann et al.[3]) are compared to each other from the perspective of a set of quality attributes (i.e. the ISO9126[7] classification). While the primary purpose of this study is not the same as for study 1 the results are such that relations can be extracted from the data set. We discuss the extraction of quality attribute relations from this study in further detail in Section 2.

Study 3 [5] is a literature survey of several sources of quality attributes and lists a large set of quality attribute relations found in different literature sources. We use the relations found in this study to compare with the industry view found in study 1 and the academic view extracted from study 2. Section 4 describes this comparison further.

The purpose of the study in this article (labelled study 4 in Figure 1) is to consolidate the industry and academic view together with the literature view. The contribution of this is to provide further confirmation of the relations between different quality attributes. In the cases where the different views disagree on a particular relation, the contribution of this article is to identify these relations as targets for further studies. For example, one explanation for disagreeing views is that a relation depends more on the specific context in which it is used than what may previously have been assumed. In this case, further studies may be necessary to fully understand the quality attribute relation.

2 Academic Study

The purpose of the academic study [14] is to test a method for assessing software architecture structures with respect to their quality attributes. The idea is that instead of trying to rate the value for quality attributes on an absolute scale, the evaluation is done relative to other architecture structures and quality attributes. A method from management science (i.e. AHP [11, 12]) is used that in a structured way obtains the subjective judgements of experienced software developers, relying on their experience and knowledge to extract the qualities of the architecture structures involved.

This study was conducted in an academic setting with participants working at the local university, most of which have considerable industry experience as well. For this study we used a subset of abstract architecture patterns from Buschmann et al.[3], and the quality attributes from ISO9126[7].

As an outcome of [14], two tables were created, rating the architecture patterns from two different perspectives. These two tables are presented in Table 2 and Table 3.

These two perspectives are such that Table 2 rates which architecture pattern that is perceived as best at supporting each quality attribute. In the original study we refer to this table as a framework for quality attributes (FQA). This table should be read row-wise. Table 3 rates the support each

architecture pattern is perceived to provide for the different quality attributes. In the original study we refer to this table as a framework for architecture structures (FAS). This table should be read column-wise.

As these tables express a view of the involved quality attributes in terms of the architecture patterns, we believe it is possible to extract quality attribute relations to a format that is comparable to the format of the other studies (e.g. the format used in Table 1).

For Table 2, we can consider each architecture pattern as one example of how well each quality attribute is supported in relation to the other architecture patterns. This means that if the ranking of architecture patterns is similar for two quality attributes, there is a possibility that these two quality attributes are related to each other. Conversely, if the rankings are strongly dissimilar, this suggest that the two compared quality attributes are mutually exclusive. Hence, a Pearson correlation coefficient for each pair-wise combination of quality attributes will show whether there quality attributes support or are in conflict with each other.

For Table 3, it is not as straightforward to compare the quality attributes, since the rankings of the quality attributes only have a meaningful interpretation in relation to the other quality attributes within the scope of a single architecture pattern. However, we may study whether quality attributes "follow" each other in different architecture patterns, i.e. if the same patterns of quality attribute rankings appear in several of the architecture patterns. This would for example mean that if a certain quality attribute is ranked highly in one architecture pattern, a conflicting quality attribute should be ranked lowly. Moreover, if the first quality attribute is ranked lowly in another architecture pattern, the second should be ranked highly. Thus, we are able to calculate and use Pearson correlation coefficients in the same way as for the FQA table.

However, since each column in Table 3 is normalised to sum up to one, there are dependencies between the rows that may influence the correlations. One way to cope with this is to first replace the actual numbers with the corresponding rank. For example, this would mean that the column for Microkernel would read (4, 2, 1, 3, 5, 6), i.e. Microkernel is perceived to support portability best, followed in turn by maintainability, efficiency, reliability, functionality, and usability. These numbers are not depending on each other in the same way as the original vector and are thus comparable over all columns. The problem with this approach is that we also lose information in the process, since the nuances of the original vector are lost, so in the subsequent analysis we use both correlations based on the original numbers and the rank data.

In order to compile the correlations described above into a format that is comparable to the other studies, we extract those correlations that are larger than a particular threshold value. For this study we use the (arbitrarily set) threshold value of 0.5. The relations larger than 0.5 (or smaller than -0.5) are listed in Table 4. The relations common to all three different views are printed in bold letters.

Table 4. All Relations from Academic Study

Quality Attribute	vs Quality Attribute	Relation / Influence		
		FQA	FAS	
			Original	Rank
Efficiency	Functionality	neg.		
Efficiency	**Usability**	**neg.**	**neg.**	**neg.**
Efficiency	Reliability		pos.	
Efficiency	Maintainability			pos.
Efficiency	Portability	pos.[a]		neg.[a]
Functionality	Usability		pos.[a]	pos.
Functionality	Reliability		neg.	neg.
Functionality	**Portability**	**neg.**[a]	**neg.**	**neg.**
Usability	**Reliability**	**pos.**[a]	**neg.**	**neg.**
Usability	Maintainability	pos.		neg.
Usability	Portability	pos.[a]		
Reliability	Maintainability	pos.		pos.
Reliability	**Portability**	**pos.**	**pos.**[a]	**pos.**
Maintainability	Portability	pos.	neg.[a]	

a. Identified after removing one outlier

3 Relations in Literature

Henningsson [5] presents a summary of quality attribute relations found in four different sources[1, 2, 9, 10]. However, this summary makes no attempt to translate the names of the quality attributes between different classifications. In order for the summary to be useful, we need to conduct this translation. Table 5 presents a translation between the different classifications. A few things are worth noting in this translation. First, Boehm et al.[1] and Bosch[2] do not provide any reference to definitions for the quality attributes, which means that our translation may be inexact. Second, ISO9126[7] is constructed as a hierarchy of quality factors, so some of the translations exist as sub-attributes in this standard. For others we have matched the descriptions of the quality attributes. Moreover, there are three quality attributes, i.e. reusability, assurance, and cost/schedule that are not part of the ISO 9126 standard. Reusability is part of McCall[10] and Boehm et al.[1], whereas the latter two are unique for Boehm et al.[1].

Using the translations in Table 5, we get a list of relations between quality attributes, presented in Table 6 and expressed in the categories of ISO9126[7], . In this table, we see the ISO 9126 equivalents where there exist one, the relations between the quality attributes, and the sources of these relations. If there is no equivalent quality attribute category in ISO9126[7] we have kept the original quality attribute, printed in italics.

As seen in this table, we are only able to confirm one relation: efficiency and functionality (where functionality has the original meanings interoperability, security, and in-

Table 5. Translation of Quality Attributes between Different Classifications

ISO 9126	McCall 1994	Other Sources
Efficiency	Efficiency	Performance [1]
		Performance [9]
		Real Time [2]
Functionality	Interoperability	Interoperability [1]
	Integrity	Security [9]
	Correctness	
Maintainability	Maintainability	
	Testability	
	Flexibility	Flexibility [2]
Portability	Portability	Portability [1]
		Evolvability [1]
Reliability	Reliability	Reliability [9, 2]
		Fault Tolerance [2]
Usability	Usability	Usability [1]
No Equivalent in ISO9126	Reusability	Reusability [1, 2]
	No Equivalent in McCall	Assurance [1]
		Cost/Schedule [1]

tegrity) is confirmed in three of the four literature references. For the remaining relations there is thus little confirming evidence.

4 Comparing Views

To summarise, the quality attribute relations found in study 1 (i.e. an industry survey [6]) are presented in Table 1. The relations found in study 2 (i.e. an academic study [14]) are presented in Table 4. The relations found in study 3 (i.e. a literature review [5]) are presented in Table 6. In this section we combine these three different views from different sources, with the intention of confirming the found relations.

The approach we propose is to first investigate the combination of study 1 (industry view) and study 2 (academia view), and then combine this with the relations found in study 3 (literature view). The relations identified in this way are presented in Table 7. Within parentheses in the literature column we see the number of literature sources that mention the particular relation.

Studying Table 7, we see that one relation, i.e. between usability and efficiency is listed as a negative relation in all three cases. This serves as a confirmation of this relation.

The negative relation between efficiency and functionality is found both in literature and in the industry study, which thus provides some confirmation of this relation. This relation is further strengthened by the fact that three literature sources mention this relation as negative.

There are three relations where the literature view is contradicted by the industry survey or by the academic study, i.e. the relations between functionality and portability, maintainability and reliability, and portability and maintainability. It may be the case that these relations are apparently not as well understood as may have been previously believed, and that further studies on these relations are needed.

Table 6. Quality Attribute Relations from Literature, Expressed in ISO9126 Terminology

Quality Attribute	vs. Quality Attribute	Relation / Influence	Source
Assurance	*Cost/Schedule*	neg.	[1]
Assurance	Efficiency	neg.	[1]
Assurance	Functionality	pos.	[1]
Assurance	Portability	neg.	[1]
Assurance	Usability	neg/pos.	[1]
Efficiency	Functionality	neg.	[1, 9, 10]
Efficiency	Portability	neg.	[1, 10]
Efficiency	Reliability	neg.	[2, 9]
Maintainability	Efficiency	neg.	[10]
Maintainability	Functionality	neg/pos.	[10]
Maintainability	Maintainability	pos.	[10]
Maintainability	Reliability	neg.	[10]
Maintainability	Usability	pos.	[10]
Functionality	*Cost/Schedule*	neg.	[1]
Functionality *(Interoperability)*	Functionality *(Integrity)*	neg.	[10]
Functionality	Portability	pos.	[1, 10]
Functionality	Usability	pos.	[1, 10]
Maintainability	Efficiency	neg.	[2, 10]
Maintainability	Functionality	pos.	[10]
Maintainability *(Testability)*	Maintainability *(Maintainability)*	pos.	[10]
Maintainability	Reliability	pos.	[10]
Maintainability	Usability	pos.	[10]
Portability	*Cost/Schedule*	neg.	[1]
Portability	Maintainability	pos.	[10]
Portability	*Reusability*	pos.	[1, 10]
Portability	Usability	neg.	[1]
Reliability	Functionality	pos.	[10]
Reusability	Efficiency	neg.	[10]
Reusability	Maintainability	pos.	[10]
Reusability	Functionality	neg.	[10]
Reusability	Maintainability	neg/pos.	[2, 10]
Reusability	Reliability	neg.	[10]
Usability	Efficiency	neg.	[10]

5 Conclusions

When developing a software system it is important to understand the relations between quality attributes. Otherwise, effort may be spent on trying to satisfy requirements on quality attributes that are inherently in conflict with each other[2]. Conversely, development may be facilitated if it is known that by satisfying one quality attribute, other quality attributes are likely to follow because of a positive relation. Hence, there is a need for an increased understanding of the relations between quality attributes.

In this article we study several sources of quality attribute relations (i.e. several literature sources, an industry

[2]This is not to say that conflicts cannot be resolved, just that another approach need to be taken, as e.g. [4] exemplifies.

Table 7. Mapping of Relations between the Three Views using ISO9126 Terminology

Quality Attribute	vs. Quality Attribute	Relation / Influence		
		Industry	Academia	Literature
Efficiency	Functionality	neg.		neg. (3)
Functionality	Portability		neg.	pos. (2)
Functionality	*Time to Market*	neg.		
Maintainability	Reliability	neg.		pos. (1)
Reliability	Usability	pos.	neg./pos.	
Reliability	Portability		pos.	
Portability	Maintainability	neg.		pos. (1)
Usability	Efficiency	neg.	neg.	neg. (1)

survey and relations extracted from a study on architecture assessments performed in academia). We triangulate these different sources against each other looking for confirming or confuting evidence of the relations found in each of the sources. As a result of this triangulation we find confirming evidence for the following:

- There is a negative relation between usability and efficiency
- There is a negative relation between functionality and efficiency

In addition, we found conflicting results for the relations between functionality and portability, maintainability and reliability, and maintainability and portability. Further studies are necessary to understand the reasons for this. For example, the specific context in which the quality attributes are studied may influence the results more than previously believed.

In conclusion, the contribution of this article is that although some quality attribute relations can be confirmed by more than one source, most relations are only presented in one source. In order to be able to use classifications of quality attributes, and in particular relations between quality attributes, it is important that the classifications can be trusted. As this article indicates, one cannot use existing classifications blindly, but must rather look at the specifics of each situation. Clearly, more evidence is needed in order to understand the relations between different quality attributes, in order to provide more accurate decision support in software development projects.

References

[1] B. Boehm and H. In. Identifying quality requirement conflicts. *IEEE Software*, 13(2):25–36, 1996.

[2] J. Bosch. *Design & Use of Software Architectures - Adopting and Evolving a Product Line Approach*. Addison-Wesley, Harlow UK, 2000.

[3] F. Buschmann, C. Jäkel, R. Meunier, H. Rohnert, and M. Stahl. *Pattern-Oriented Software Architecture - A System of Patterns*. John Wiley & Sons, Chichester UK, 1996.

[4] D. Häggander, P. Bengtsson, J. Bosch, and L. Lundberg. Maintainability myths causes performance problems in parallel applications. In *Proceedings of the 3rd IASTED International Conference on Software Engineering and Applications*, Anaheim CA, 1999. IASTED/ACTA Press.

[5] K. Henningsson. Trade-offs and conflicts between quality attributes. Master's thesis, Blekinge Institute of Technology, 2001.

[6] K. Henningsson and C. Wohlin. Understanding the relations between software quality attributes - a survey approach. In *Proceedings of the 12th International Conference for Software Quality*, 2002.

[7] Software qualities iso/iec fdis 9126-1:2000(e)., 2000.

[8] E. Johansson, M. Höst, A. Wesslén, and L. Bratthall. The importance of quality requirements in software platform development - a survey". In *Proceedings of HICSS-34*, Los Alamitos CA, 2001. IEEE Computer Society.

[9] G. Kotonya and I. Sommerville. *Requirements Engineering — Processes and Techniques*. John Wiley & Sons, Chichester UK, 1998.

[10] J. McCall. *Encyclopædia of Software Engineering*, chapter Quality Factors, pages 958–969. John Wiley & Sons, 1994.

[11] T. L. Saaty. *The Analytic Hierarchy Process*. McGraw Hill, Inc., New York NY, 1980.

[12] T. L. Saaty and L. G. Vargas. *Models, Methods, Concepts & Applications of the Analytic Hierarchy Process*. Kluwer Academic Publisher, Dordrecht the Netherlands, 2001.

[13] M. Svahnberg. An industrial study on building consensus around software architectures and quality attributes. *Journal of Information and Software Technology*, 46(12):805–818, 2004.

[14] M. Svahnberg and C. Wohlin. An investigation of a method for identifying a software architecture candidate with respect to quality attributes. *Empirical Software Engineering - an International Journal*, 10(2):149–181, 2005.