JYVÄSKYLÄ STUDIES IN COMPUTING
149

Auvo Finne

TANZANIT
Towards a Comprehensive Quality Meta-Model for Information Systems

Case Studies of Information System Quality Modelling in East Africa

JYVÄSKYLÄN YLIOPISTO

JYVÄSKYLÄ STUDIES IN COMPUTING 149

Auvo Finne

TANZANIT
Towards a Comprehensive Quality Meta-Model for Information Systems

Case Studies of Information System Quality Modelling in East Africa

Esitetään Jyväskylän yliopiston informaatioteknologian tiedekunnan suostumuksella
julkisesti tarkastettavaksi yliopiston Agora-rakennuksen salissa AgAud3
joulukuun 20. päivänä 2011 kello 12.

Academic dissertation to be publicly discussed, by permission of
the Faculty of Information Technology of the University of Jyväskylä,
in building Agora, Auditorium 3, on December 20, 2011 at 12 o'clock noon.

UNIVERSITY OF JYVÄSKYLÄ

JYVÄSKYLÄ 2011

TANZANIT
Towards a Comprehensive Quality Meta-Model for Information Systems

Case Studies of Information System Quality Modelling in East Africa

JYVÄSKYLÄ STUDIES IN COMPUTING 149

Auvo Finne

TANZANIT
Towards a Comprehensive Quality Meta-Model for Information Systems

Case Studies of Information System Quality Modelling in East Africa

UNIVERSITY OF JYVÄSKYLÄ

JYVÄSKYLÄ 2011

Editors
Seppo Puuronen
Department of Computer Science and Information Systems, University of Jyväskylä
Pekka Olsbo, Harri Hirvi
Publishing Unit, University Library of Jyväskylä

URN:ISBN:978-951-39-4600-5
ISBN 978-951-39-4600-5 (PDF)

ISBN 978-951-39-4599-2 (nid.)
ISSN 1456-5390

Jyväskylä University Printing House, Jyväskylä 2011

ABSTRACT

Finne, Auvo
Tanzanit - Towards a Comprehensive Quality Meta-Model for Information Systems: Case Studies of Information System Quality Modelling in East Africa
Jyväskylä: University of Jyväskylä, 2011, 209 p.
(Jyväskylä Studies in Computing
ISSN 1456-5390; 149)
ISBN 978-951-39-4599-2 (nid.)
ISBN 978-951-39-4600-5 (PDF)
Finnish summary
Diss.

The main goal of the study is to overcome the problems of traditional quality models by developing a higher level conceptual framework, called the quality meta-model, to assist in modelling quality of information systems as end products of the development process. The aim is also refine the definition of information system quality and quality requirements. As a by-product the study presents principles of using the proposed meta-model as a governing element in system development process. The main research question concerns the validity of the conceptual framework. The approach is a combination of conceptual analysis, theory-creating and theory-evaluating research. In addition to an initial literature review and references to relevant writings, the research activity comprises three quality modelling case studies in East-Africa. Constructs found through a selective literature review are first generalized and combined into a conceptual framework. Thereafter this initial meta-model is used in case studies as a template for system specific quality models. Based on findings and further literature review the conceptual framework is eventually developed into its final form. The study shows that a comprehensive meta-model, needed to fully account for information system quality, is a hybrid model that in addition to the core quality elements includes parts from the development process, system and context models. The quality of an information system is in the first place determined by the existence and intensity of certain desired relationships, or state of affairs, between the system and its context. These requirements have priority over others that can be derived from the former. A system specific quality model is the driving force of development. The three case studies support the general validity of the proposed quality meta-model through the usefulness and comprehensibility of the quality models created on basis of it. It proved also to be general and flexible enough to be applied in three different contexts. Comparison to traditional quality models and newer alternatives, in turn, proves the meta-model's comprehensiveness.

Keywords: quality meta-model, quality model, quality attribute, quality design, quality-driven development, requirement

Author's address Auvo Finne
Pitkäkatu 44
Jyväskylä, FIN-40700
Email: auvo.finne@jyu.fi

Supervisors Professor Seppo Puuronen
Department of Computer Science and Information Systems
University of Jyväskylä, Finland

Doctor Anja Mursu (2005-2007)
Department of Computer Science
University of Kuopio, Finland

Reviewers Professor Gaëlle Calvary
Equipe Ingénierie de l'Interaction Homme-Machine
Laboratoire d'Informatique de Grenoble, France

Assistant Professor Eleni Berki
Faculty of Information Technology
University of Tampere, Finland

Opponent Docent Mikko Korpela
School of Computing
University of Eastern Finland

ACKNOWLEDGEMENTS

The study got its start about ten years ago when I used to travel in Tanzania and met my long time friend Mr. Anders Lönnqvist in Dar es Salaam. He was working as a consultant at the Ministry of Education and Culture. For my part I was at that time considering to resume my studies and I was looking for a research object. Mr. Lönnqvist introduced me to some employees at the ministry who were involved in an Education Management Information System (EMIS) project. In the meeting I suggested adding a quality ensuring component to the project and started applying for research clearance and permit.

In Finland, at my university in Jyväskylä, I was happy to meet with and present my research plan to Professor Seppo Puuronen, who responded positively and became my supervisor and later series editor. I'm grateful to him for continuous support through years in applying funds and guiding my work up to details of editing and organizing the thesis. I was also lucky to get Dr. Anja Mursu as my co-supervisor in the early stages of the research. Based on her previous work in Africa she could share her contacts and give valuable input to the process. A third supporting factor in Finland was that I gained admittance to and funding in several installments from the Jyväskylä Graduate School in Computing and Mathematical Sciences (COMAS).

Next, I want to thank the reviewers of my thesis, Professor Gaëlle Calvary and Doctor Eleni Berki, for their clear and constructive feedback. It helped me to make some additions to the manuscript and especially it has given me ideas for future research and development of the quality meta-model.

On the Tanzanian side I'm grateful to Mr. Cyprian Miyedu, Mr. Godwin Noah, Mr. Godfrey Ponera and Mrs. Assela Luena to finally get a warm welcome and a status as EMIS consultant and researcher in the team. In addition I would like to thank officers at COSTECH (Tanzania Commission for Science and Technology), professor B. Mutagahywa, my local contact, and professor V. Fournadjiev at the University of Dar es Salaam for assistance in evaluating my work and getting a yearly research permit in Tanzania. And I cannot forget all the support from every employee at the ministry's statistical unit, personnel at the Finnish Embassy and my personal friend and assistant in Dar es Salaam Mr. Kennedy Ndaro.

For the second case study opportunity I'm grateful to two ladies Saara Ojanen and Maija Hakulinen at Finn Church Aid who sent me to work in Mozambique on a web site project. I would also like to thank country representative Philip Wijmans, program coordinator Lucas Owuor-Omondi, assistant administrator Titos Languene and all others at LWF (Lutheran World Federation) Maputo office for intensive cooperation.

For the third case study opportunity I'm grateful to Mr. Jukka Nieminen and Dr.Tuomo Heinonen, chief technical advisors of the SMOLE (Sustainable Management of Land and Environment) project in Zanzibar. Both advisors had confidence in my ability to work in an Islamic context and I got a rare context to produce a final version of information system quality meta-model. I would also like to thank the directors departments at the Ministry of Water Construction Environment and Lands (MWCEL) Mr. Haji A. Haji, Mr. Said O. Fakih and Mr. A. Juma, as well as all the employees in these departments. My special thanks go to Mr. Pertti Onkalo, land management advisor, GIS advisors Mr. Pekka Hurskainen, Mr. Arto Vuorela, Miss Hanne Junnilainen, and all other colleagues on that tropical island.

In the course of the research activity I was given the opportunity to liaise with two viable researcher networks, the INDEHELA (Informatics Development for health in Africa) project lead by Professor Mikko Korpela and the Information and Communication Technology Education for Development project lead by Professor Erkki Sutinen. I would like to thank all members of these two groups. In addition I would like to thank Professor Juhani Koponen and all the members of our unofficial researcher group in Dar es Salaam for fruitful interdisciplinary discussions.

Finally, I want to thank my wife, other family members and friends for having patience with me and my research work. On one hand it has given them insights into African culture, but on the other hand it has meant giving up some own needs and long periods without a husband and father in the house. Thanks go also to my both parents and, with respect to scientific view of life, especially to my father and his family tradition.

Kigoma, Gombe National Park, Tanzania 25 November 2011

Auvo Finne

FIGURES

TABLES

CONTENTS

1 INTRODUCTION

After decades of information and communication technology use in industrialized countries, there still exists a wide dissatisfaction among users with the systems they are tied to in their everyday work. In 2004 one software business executive summarized that “quality is probably the most important element of this decade because there are still many problems in the software products that go out the door, and we all pay a huge price for that” (Kinreich 2004, 117). Georgiadou (2003) writes about “social loss”, financial and other, that is caused by low product quality. As we have entered the second decade of the new century, the problem of quality persists. In developing countries and other regions where technical and financial resources are scarce, the quality of information systems is a truly critical issue.

There are many reasons behind the persistence of quality problems. First, the word “quality” is fashionable and often overused. And like in case of other buzzwords, the underlying concept itself is often vague and fuzzy. The problem of definition is noted for example by Nelson, Todd & Wixom (2005), Côté, Suryn & Georgiadou (2007) and Scholl, Eisenberg, Dirks & Carlson (2011). It is often difficult to make clear what “quality” means and relate quality requirements to other requirements. Second, a number of problems are associated with traditional and prevalent quality models (e.g. McCall, Richards & Walters (1977), Boehm, Brown, Kaspar, Lipow, McCleod & Merrit (1978), Dromey (1996), even ISO 9126 (2001). They are felt to be prescriptive and to cause too much overhead to development work in terms of calendar time, money and other resources that are badly needed to address pure technical challenges. For that reason these models are in practice easily neglected. ISO 9126 even itself admits that it is not possible to measure all characteristics of a large software product or to measure them for all scenarios. Miyoshi & Azuma (1993) point out that the number of quality characteristics should be kept between three and eight for cognitive and practical reasons even if this is not enough to define quality in detail. Further, despite their prescriptiveness, traditional models are incomplete and biased towards metrics. These models usually have two main parts, i.e. a categorized collection of quality attributes and a system for measuring attribute values in practice. On the one hand such collections help to learn about different quality characteristics, their general definitions and some logical relationships, and the collections can be used to disclose gaps in the general quality design of an information system. The metrics part, in turn, can give detailed instructions on how to measure the presence and intensity of individual qualities. But on the other hand categorized attribute lists do not tell all about quality and do not suffice to model and evaluate either overall system quality or individual quality attributes comprehensively. In other words, the traditional quality models don't cover all necessary aspects of quality or quality modelling in a clear, comprehensive and balanced manner.

Despite of restrictions in traditional quality models, an examination of articles about IS quality reveals that it is still difficult to find exhaustive discussion on alternative overall frameworks for quality modelling and on the ontology behind the models. Individual qualities, instead, like usability, accessibility, performance, security, modifiability, etc. are discussed regularly. Some initiatives in creating a theory of quality models can, however, be found. One of them is the systemic quality model (SQMO) developed in 2001 at the Universidad Simon Bolivar in Venezuela and used for example by Ortega, Pérez & Rojas (2003) who discuss previous attempts to build a quality model and present their own alternative, which they term a systemic quality model. The term systemic means that the model covers both the development process and the end product. Other models have also been proposed. Georgiadou (2003) presents a graphical model and method of visualizing and quantifying different stakeholder views of product quality. Wong (2004) puts forward a software evaluation framework that relates software characteristics to stakeholders, desired consequences and sought after values. A system type specific quality model, in turn, has been proposed by Jureta, Herssens & Faulkner (2009). It targets service oriented systems and refers explicitly to the ontology of quality. When it comes to individual quality attributes, a method of modelling them conceptually has been presented, for example by Cysneiros & Leite (2004). Côté et al. (2007) go somewhat into the direction of ontology, proposing three requirements that a quality model should possess. Finally, a quality meta-model called QUIMERA (QUality metamodel to IMprove the dEsign RAtionale) has recently been put forward by Frey, Céret, Dupuy-Chessa & Cavalry (2011).

To address the problems mentioned above and to contribute to the discussion about information system quality and quality models this study presents a general conceptual framework as an alternative for modelling product quality called the "information system quality meta-model", a model or template for actual quality models. Further, based on the meta-model a refined definition of information system quality, quality requirements and quality attributes is presented. In other words the study looks at the ontology of the actual models. The meta-model focuses more on how to specify quality than on how to implement and measure it. The two latter activities are based on actual quality models and constitute mandatory elements in information system development process. As a secondary outcome the study suggests how to use the meta-model and demonstrates how quality modelling can in fact lead the entire system development process.

The proposed meta-model shows what aspects need to be understood, described or designed in order to account comprehensively for the quality of a particular information system as an intermediate or final product of development process. It views an information system as a technical artefact or tool in its context of development and use, and provides conceptual means for defining quality attributes, termed "-ilities" by Voas (2004). It identifies the parts and the overall structure of actual system specific quality models as well as of individual attribute models, but it does not go into details of requirements engineering technologies, modelling languages or presentation techniques. A meta-level model does not put a similar burden and a similar set of prescriptions on developers as the traditional lower level quality models. It ensures, however, that even if only a certain number of qualities are given attention, they are handled properly taking into account all necessary aspects. Accordingly it is also likely to be more applicable in different cultural and infrastructural contexts than the lower level models.

The main research question concerns the creation and validity of the conceptual framework or meta-model. The approach in this study is a combination of conceptual analysis, theory-creating and theory-evaluating research (presented in detail by Järvinen (2001)). In order to find all essential perspectives, aspects and factors needed to fully understand information system quality and to put these pieces together to form a general and comprehensive conceptual framework for modelling the quality of an information system as an end product of development process, the following research activities were carried out: 1) an initial literature review, 2) conceptual analysis-synthesis resulting in a first version of meta-model, 3) a series of three quality modelling case studies in East-Africa, under the name TANZANIT for testing applicability of the framework, 4) analysis of results and creation of the final model and 5) a complementary literature review comparing the findings to most recent writings. The meta-model was initially generalized and synthesized from a set of selected writings, among them the ISO 9126 (2001) quality model, and researcher's own experience in system development. Behind ISO 9126 (2001), in turn, lie some of the commonly known traditional quality models mentioned above. These are accordingly included in the chapter about theoretical background (cf. ISO 9126 2001, 23). The study then examines the process of using the framework to create system specific quality models for real systems under development and uses the implications to modify the framework and to increase its validity. The approach used has some features that are similar to action research and grounded theory. The two first case studies are also described in separate articles written by the author (Finne 2006 and 2011).

The study contributes to the research in the field of information system quality modelling by giving quality a definition that separates it logically from requirements in general and by combining essential aspects of product quality modelling into one comprehensive conceptual framework. In addition, it gives practical examples of quality modelling using the meta-model. The examples are of special interest to developers who need to work in the infrastructural and cultural context of Africa. Finally, the study addresses the needs of system developers by suggesting how the meta-model can be used in practice and how the resulting quality model can drive the whole development process. This method, combined with the proposed understanding of quality and quality attributes, could be called "quality-driven development" in contrast to approaches like "test driven", "use case driven", etc.

The study has a number of limitations related mainly to the fact that it is based in substantial part on case studies. This method has its well known problems of researcher's influence, replicability and generalizability. Only three case studies were carried out and all within East-Africa. In addition none of the cases had a software company as the primary user of the meta-model, and the quality models created during case studies were more like prototypes than full blown detailed quality models. The latter is due partly to the focus on meta-level and partly to limited time resources. Finally, the research design did not include any parallel tests with an alternative quality model, which may be viewed as a limitation.

The thesis consists of 8 chapters following the order of activities in the research process. The first chapters lay the groundwork for the study by introducing the topic, explaining its importance and describing the theoretical background and framework (Chapter 2). They also set the goals for the study. Chapter 3 opens the research questions and discusses the research process, its structure and methods. First limitations of traditional quality models are described. Then overall research design is presented together with detailed research questions. Finally the chapter describes the design of case studies. Chapter 4 presents the initial information system quality meta-model that is used in the first two case studies. It is described element by element and finally compared to the theoretical background. Chapter 5 presents in detail the three case studies. It starts with a general description of Tanzania and Mozambique as information societies. Next the first two cases are treated together because even in practice they were intertwined. Before description of the last case study an intermediate meta-model is brought up and explained how it grew out of the preceding case studies. The description of case studies is organized according to the meta-model, presenting the application element by element. In addition the concept of information system quality is discussed in the light of case studies. Chapter 6 brings together the findings of all three case studies. What is discovered is first used to answer the research questions. The last section of the chapter suggests how the meta-model can be used in practice. It also discusses the role of quality model in system development process. Chapter 7 presents the final version of information system quality meta-model. In addition to an overall view, each element is given a more advanced definition compared to the initial version. Concept of quality is also defined and the nature of system specific quality models shortly discussed. The last section evaluates the proposed meta-model by comparing it to the more recent discussion and alternatives. Finally, in Chapter 8 the study, its results and contributions are summarized, and directions for future research are presented.

2 THEORETICAL BACKGROUND

This chapter discusses different types and levels of models and modeling associated with information systems. Next it defines the characteristics of the model which is the target of this study. These steps are part of the conceptual analysis. Next the chapter presents a concise literature review of the most well known traditional quality models and some additional theories that constitute the background for the first version of quality meta-model. Discussion of how the relevant constructs found in the background models and theories are related to the meta-model is left to chapter 4's presentation of the initial meta-model.

2.1 Quality modelling, models and modelling levels

A human made information system (IS) exists and operates in the context of societies, organizations and personal lives. The expression "information system" is here, and throughout the study, used in the sense of computer system, i.e. a mechanical (today usually digital) tool used to assist in dealing with information. In all cases information should have meaningful functions, and because of these functions human actors have various expectations about a system's behavior and other features. This wider view is implicit for example in writings where sense-making theory is applied to information system design (e.g. Muhren, van den Eede & van de Walle (2008)). Sense-making provides means to understand how information in general is used by humans (see e.g. Savolainen (2006)), which in turn should be reflected in information tools design. The general use of information, a kind of "wrapper" around technology, is also called "information behavior" in theoretical papers (e.g. Allen, Karanasios & Slavova (2011)). The information system supports the information behavior, which in turn supports the human activity. Another theory frequently used (e.g. Silva 2007, Macome 2008) to explain the interplay between system and context is Actor Network Theory (ATN). ATN views technology as part of a network of human actors and nonhuman artifacts (Macome 2008, 157). Finally, the above ideas are also visible in Alter's (1999, 2008) definitions of work system and information system. The study at hand does not, however, further rely on these theories in the analysis.

In everyday life and scientific literature actors' expectations are commonly called requirements, and a part of them more precisely quality requirements. Requirements set by human actors must be differentiated from requirements that are based on laws of nature or logic. Each IS has a life course that contains various cyclical or repeating processes, like system development, moving from one platform to another, etc. ISO 12207 (2008), for example, gives a good picture of software lifecycle processes. In the course of these processes there are several points where requirements are captured and goals set, something – usually a system or component - to fulfill the requirements modeled, implemented and put into operation, and finally the results measured and evaluated. A plethora of technologies have been developed to assist in the work. Parviainen & Tihinen (2007) present a good overview of them.

The words "modeling" and "designing" are used in this study interchangeably. And a "model" in this connection means the product of modelling or designing process, an abstraction of or a blueprint for something to be realized (cf. Jacobson, Booch & Rumbaugh 1999, 22). It describes the target on a quite practical level. The difference between requirements and a model based on them is that requirements express needs and desires and are often less structured and consequently written in the form "x is needed" or "x must be y", whereas a model is structured and statements are in indicative form like "x is y". During information system life cycle one can set requirements for and create models of different things. The most common objects are the system itself as a product of development, the development process and the entire system life course. But requirements can be set even for the models themselves. The above understanding of contexts, processes and models is depicted in FIGURE 1. The dotted ellipse depicts information as part of human activity and interaction. The former serves the latter. Information systems as tools, in turn, must provide some added value in this two-tiered context.

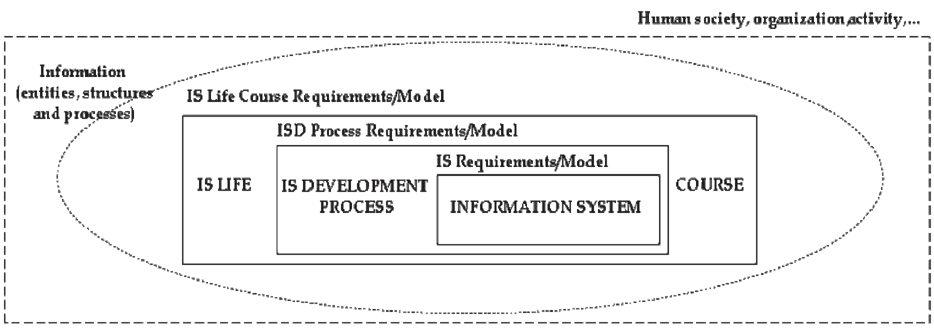


FIGURE 1 Information system - contexts, processes and models

Being aware of the connection between the system development process and the resulting system, this *study focuses on modeling and models that are based on quality requirements* and set *for an information system in its context of use as an artifact, an intermediate or end product* of development process. Consequently, here "information system quality model" means *a model of or a blueprint for the constituents of quality of a product in its context of development and use*. Only when discussing the general features of the quality modeling process itself the study steps on the side of process modeling.

Modeling can take place on different levels of abstraction. And the object (X) that is modelled can be any real thing, not only an entity or process, but even *a state of affairs* as this study will show. Often the designer has a requirements document at hand, but this is not necessary as in case of prototyping. After modelling and implementing several Xs one can create a general model of Xs or of certain type of Xs that consequently constitute theories of X. Finally, instance and general level models can be used to create a meta-model, a model for X-models that can be regarded as an even higher level of theory. The other way around, higher level model can be used as template for creating lower level models. By iterating this kind of model generation, from bottom up and from top down, models on all levels grow better and better by the time and experience. FIGURE 2 depicts this relationship between levels of models, and at the same time between practice and theory.

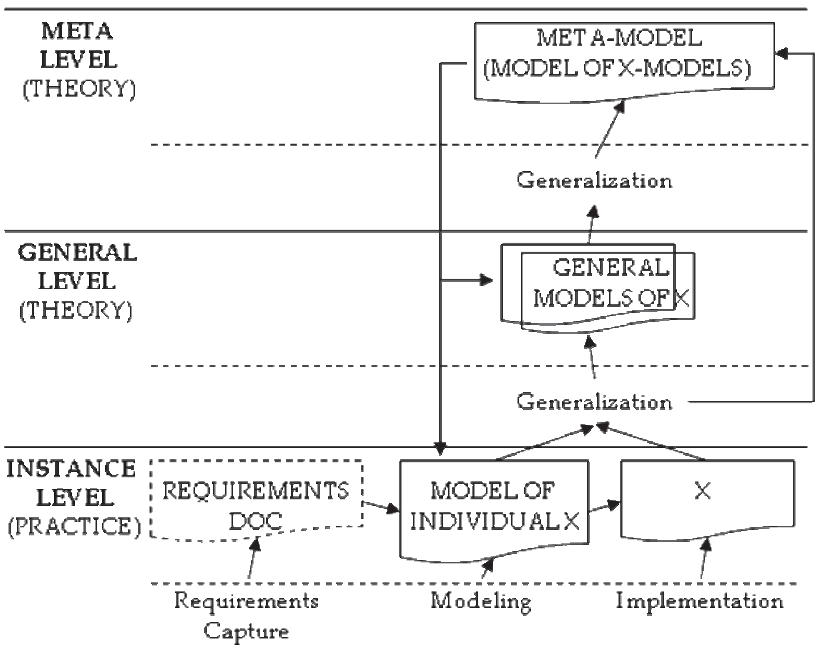


FIGURE 2 Levels of abstraction in modelling – practice and theory

Accordingly, quality modelling can take place on three different levels (FIGURE 3). Discussion about what *quality, quality attribute, quality model*, each *model* *element* or *aspect*, etc. mean, belongs to the highest level, prefixed with "meta". The lowest level, in turn, can be called the instance-level equalling blueprint for the system with respect to what is described. All system and project specific considerations and descriptions can be found on this level. Examples of quality modelling and quality measurement described in chapter 5 give an account of case studies belonging to the instance level. Finally, the objects of the middle-level are formed by generalizing from information gathered about instances of quality modelling in connection with different information system projects in different contexts. In other words, instance-level represents empirical evidence and general and meta-levels represent theory. The quality models and theories presented below in the initial literature review belong mainly to the middle level. They assert something general about qualities or quality measurements, but don't go further in the abstraction. This *study*, in turn, as will be described in following chapters, *focuses on the* rarely discussed *meta-level*. It takes input, however, from general level theories and instance level case studies.

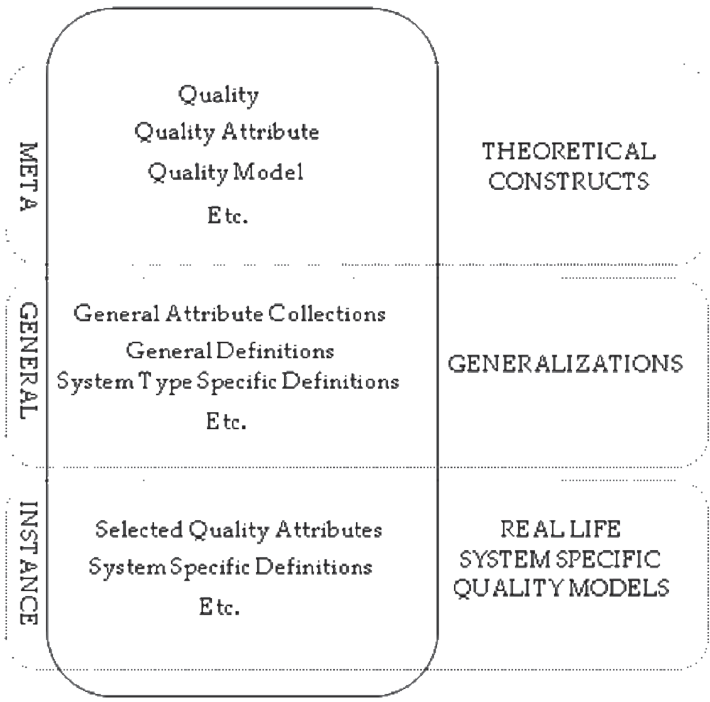


FIGURE 3 Levels of abstraction in quality modelling

The middle level of quality modelling can accommodate many different types and degrees of generalization. The possibilities are endless. General attribute collections can reflect good characteristics common, for example, to all statistical systems, websites, hard disks, routers or user interfaces respectively. Nielsen (1993, 91), for example, notes two of the modelling levels when he discusses guidelines for user interface design. He differentiates between general (all user interfaces), category-specific (user interface for a certain type of system) and product-specific guidelines. In addition to technical categories, generalizations can be made according to system environments, taking into account both infrastructure and other circumstances. What is common, for instance, to information system safety in an industrialized country like Finland might not be applicable as such in East-Africa. And standards like ISO 9126 are suspected of primarily reflecting conditions in high-tech societies. This difference is very well noted and described for example by van Reijswoud (2009). In principle, creating a quality model for a particular cultural, infrastructural and economical context would mean starting fresh from the meta-level, selecting the most important quality attributes, defining them with respect to the context in question, etc.

2.2 A literature review

The initial information system quality meta-model for the case studies was first drafted on basis of a collection of writings from software business executives (Kearns & Falls 2004), Nielsen's (1993) usability model, the ISO 9126 standard, a handbook of object oriented software development written by Jacobson, Booch and Rumbaugh (1999) and articles about architecture analysis (e.g. Svahnberg & Wohlin C. (2005)). This selection offers a variety of views on product quality ranging from business people, to a widely used development handbook, to information system research, and finally international standards. ISO 9126 (2001), the latest standard overall quality model, was preceded by similar frameworks of which the best known are McCall et al. (1977), Boehm et al. (1978), Grady and Caswell (1987) and Dromey (1996). Accordingly, all the above mentioned works can in addition be counted to the theoretical background of the study. The main features of these writings are presented below, in a chronological order, except that the "non-theoretical" view held by business executives is discussed first.

2.2.1 The non-theoretical view

A good starting point for studying information system qualities is to examine how these appear in the minds of software business actors who are neither software engineers nor researchers. For example a collection of writings (edited by Kearns & Falls (2004)) from software business executives about important issues for software success can provide such material. TABLE 1 lists the characteristics (some exact expressions used by writers may differ from those used in the table), which can be found in the articles. The authors use adjectives, verbs or abstract nouns to describe qualities of software and information systems, resulting in the application of natural language for referring to qualities. In the table exact expressions appearing in the articles have been synthesized into 35 categories each representing one characteristic or a group of related characteristics. The collection of characteristics is heterogeneous and the meanings of some adjectives are overlapping. For example in the second table row meanings of words "'adaptable", "adaptive", "modifiable", "customizable" and "scalable" are very closely related. The numbers in the column on the right side indicate, how many of the software business executives mention the property in question. It can be taken as a simple ranking of the properties. It must be kept in mind that the essays in the book were written about the most important issues for software industry's future success, not directly about the most important qualities of software.

TABLE 1 Characteristics of a good product in the minds of software business executives

Category	Characteristic Name	Number of Authors
1	supported	9
2	adaptable, adaptive, modifiable, customisable, scalable	9
3	needed	8
4	compatible, integrated, standardized	7
5	(cost-)efficient, productive (for customer)	7
6	improving, innovative, evolutionary	7
7	useful, helpful, used	6
8	desirable, wanted	5
9	modern, new	4
10	quality	3
11	easy to use	3
12	unique	3
13	entertaining, exciting	3
14	effective, influential	2
15	functioning correctly	2
16	secure	2
17	simple	2
18	generalized	2
19	easy to deliver	2
20	profitable (for producer)	2
21	acceptable	1
22	available (without downtime)	1
23	intelligent	1
24	relevant (to market)	1
25	correct (data integrity)	1
26	tested	1
27	easy to learn	1
28	easy to install	1
29	comprehensive	1
30	well-architected	1
31	component based	1
32	portable	1
33	high performance	1
34	profitable (for customer)	1
35	up to date	1

The listing in TABLE 1 shows that people are commonly well aware of the many characteristics of good software. In the minds of software business executives the highest ranked properties of software are that it is supported, adaptable and customisable and that it fills the needs of customers. The writers of the essays do not describe, however, how these good properties can be conceptualised, taken care of and measured. Neither definitions nor categorizations can be found in the book, just ordinary sentences containing the adjectives listed in the table. This way of dealing with information system qualities can be called the "non-theoretical" or "common" view of information system qualities. It is often the view of customers, end users, managers etc.

2.2.2 McCall's factors in software quality

McCall et al (1977) identified and analyzed over 50 candidate quality factors and grouped them into sets. The most descriptive factor (the actual factor) in each group was given a definition and used as group name. Reliability, for example, was defined as "extent to which a program can be expected to perform its intended function with required precision". TABLE 2 shows the final version of the model with 11 actual factors. Items within each group are seen as synonyms or as criteria for the factor.

TABLE 2 McCall's quality factors (adapted from McCall et al. 1977, 2-7)

CORRECTNESS	**RELIABILITY**	**EFFICIENCY**	**INTEGRITY**
Acceptability Completeness Consistency Expression Validity Performance	Availability Accuracy Robustness Precision Tolerance		Security Privacy
USABILITY	**MAINTAINABILITY**	**TESTABILITY**	**FLEXIBILITY**
Operability Human Factors Communicativeness Convertibility	Stability Manageability Conciseness Repairability Serviceability	Accountability	Adaptability Extensibility Accessibility Expandability Augmentability Modifiability
PORTABILITY	**REUSABILITY**	**INTEROPERABILITY**	
Transferability Compatibility	Generality Utility		

Further, McCall et al. (1977) grouped the factors according to orientations that one could take in looking at a software product, namely operation, revision and transition (TABLE 3). The latter covered such things as using the software on another type of machine, reusing parts of it and interfacing with another system.

TABLE 3 McCall's quality factor sets and orientations (adapted from McCall et al. 1977, 3-2)

PRODUCT OPERATION	PRODUCT REVISION	PRODUCT TRANSITION
Correctness	Maintainability	Portability
Reliability	Flexibility	Reusability
Efficiency	Testability	Interoperability
Integrity		
Usability		

In addition to factors, McCall et al. (1977) presented the concept of a criterion as an attribute of software or software production process by which the factor can be judged. A single criterion can affect different factors. The criteria are in one-to-one relationship with metrics. The criteria for reliability, for example, are error tolerance, consistency, accuracy and simplicity. Each criterion was given a definition. Accuracy, for example, was defined as "those attributes of the software that provide the required precision in calculations and outputs". Sub-criteria can be established. Criteria were also used as basis for assessing relationships between factors. If the same criteria have positive impact on two factors, the relationship of those factors is positive and vice versa (tradeoff).

McCall's model also included metrics, i.e. theory about how the values of quality attributes are measured. According to the model quality factors can be measured already during development process. Metrics provide a measure of criteria.

2.2.3 Boehm's software quality characteristics

Boehm et al. (1978) published a study that focused on source code (FORTRAN) metrics. They presented a set of non-overlapping quality characteristics arranged in a three-level (actually four-level) hierarchical tree structure based on implication relationships between the elements (FIGURE 4). From another viewpoint lower level characteristics in the tree can be seen as aspects of or conditions for higher level characteristics. The lowest level (not shown in figure) provides the most primitive characteristics which are used as basis for defining metrics. An individual primitive can be an aspect of different higher level characteristics. Self-descriptiveness, for example is an aspect of testability and understandability. The highest levels represent things that one wants to do with software: use it as-is, maintain it or apply it to a new environment. General utility refers to both usage and modification of a software product.

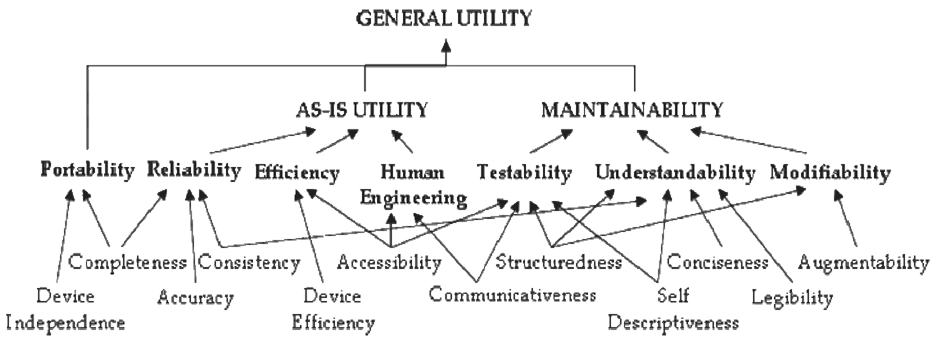


FIGURE 4 Boehm's quality characteristics (adapted from Boehm et al. 1978, 1-19)

Each characteristic was given a short definition. Understandability, for example, was defined as follows: "A software product possesses the characteristic understandability to the extent that the purpose of the product is clear to the evaluator". The definitions were followed by examples and additional detailed explanation of the characteristic in question.

The requirements-properties matrix presented in software production guidelines section is a means to relate the quality characteristics to what are called "functional requirements". 'Maintainability', for example is related to 'machine independence'. And the latter, in turn, is related in a detailed design specification list to instruction 'use standard FORTRAN'. 'Reliability' is related to 'multisection capability' which is related to design instruction 'provide mechanism for determining when there are no more sections to be processed'. (Boehm et al. 1978, 5-4, 5) These matrixes and lists can be quite extensive and they relate the quality characteristics to design elements and through them to system internals.

On the metrics side Boehm et al. (1978) developed a set of questions with respect to each characteristic for judging the quality of software product. In addition they created detailed algorithms for answering the questions. According to Boehm et al. the overall software quality is a function of the values measured for individual characteristics. It is, however, clearly stated that no single metrics can give universal ratings.

2.2.4 Hewlett-Packard's FURPS model

The model is based on work and experiences within Hewlett-Packard. For Grady & Caswell (1987) software metrics is in first place about measuring the development process with respect to factors that affect it like size of the software, cost and time for people doing the development, defects in the software, difficulty (including product complexity) of the project and amount of communication needed between stakeholders. They base their work on the principle that productivity and other gains follow naturally from quality improvements in the process. The main concern with respect to product quality is the number and type of defects experienced by the customers. Defects are categorized, for example into user interface, programming and operating environment related defects. When it comes to the final success and objectives of a development project, Grady and Caswell take up quality attribute lists like published by Boehm et al. (1978). They prefer, however a HP model called FURPS (Functionality, Usability, Reliability, Performance, Supportability). TABLE 4 shows this attribute list.

TABLE 4 Hewlett-Packard's FURPS quality attributes and groups (adapted from Grady & Caswell 1987, 159)

FUNCTIONALITY	**RELIABILITY**	**SUPPORTABILITY**
Feature Set	Failure Frequency	Testability
Capabilities	Recoverability	Extensibility
Generality	Predictability	Adaptability
Security	Accuracy	Maintainability
	Failure Mean Time	Compatibility
		Configurability
USABILITY	**PERFORMANCE**	Serviceability
		Installability
Human Factors	Speed	Localizability
Aesthetics	Efficiency	
Consistency	Resource Consumption	
Documentation	Throughput	
	Response Time	

In the table "Frequency/severity of failure" is abbreviated to "Failure Frequency" and "Mean time to failure" to "Failure Mean Time". Application of FURPS involves two steps: prioritizing attributes and making them measurable, i.e. choosing measurable goals for each attribute.

2.2.5 Nielsen's usability model

Nielsen (1993) presents a hierarchy of attributes with 'system acceptability' on top. After presenting the overall structure Nielsen's work performs a deep analysis of a single composite property, namely usability, and deals with user interface design and testing process in general. Therefore, Nielsen's model can be viewed as a kind of attribute specific model. FIGURE 5 shows how usability is positioned among other attributes of system acceptability. Groupings in the hierarchy are called "categories". Components of usability, for example, belong together because they refer to those system aspects with which a human might interact.

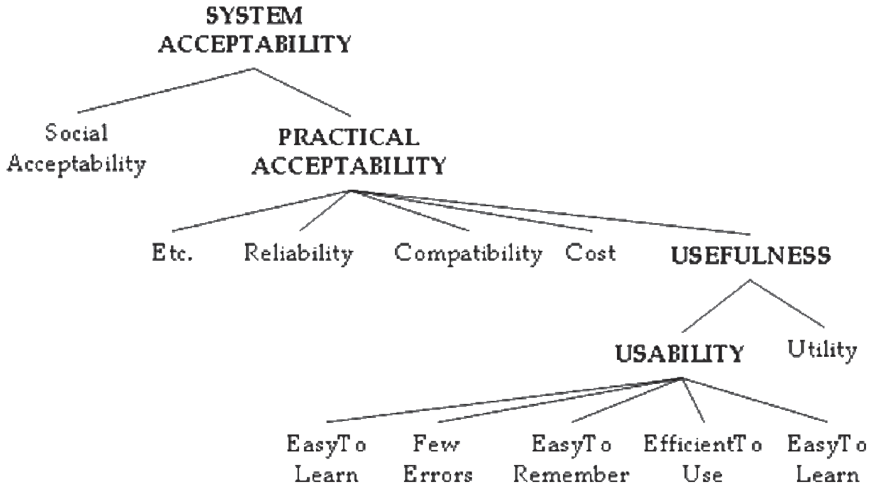


FIGURE 5 Nielsen's model of attributes of system acceptability (adapted from Nielsen 1993, 25)

Nielsen gives general definitions to usability and its components. Learnability, for example, means that "the system should be easy to learn so that the user can rapidly start getting some work done with the system". Nielsen discusses each component and its measurement extensively. The hierarchical model relates attributes as components and combinations. Nielsen notes also the trade-off relationship and the need for prioritization and setting goals in terms of measured usability (Nielsen 1993, 25, 41, 80). Components of usability are measurable and are measured relative to specific users and tasks. System's overall usability, in turn, can be determined on the basis of the mean values of components together with the distribution of these values. Nielsen differentiates between three major categories of users: novices, experts and casual users. In addition, the experience differs along three dimensions: experience with the system, with computers in general and with the task. Further factors for grouping users are: age, gender, reasoning abilities, etc. According to Nielsen the concept of 'user' should include everybody whose work is affected by the information system (Nielsen 1993, 73).

2.2.6 Dromey's quality model framework

According to Dromey (1996, 34) a "product's tangible internal characteristics or properties determine its external quality attributes". Linking internal quality-carrying properties to quality attributes is, however, not absolute. Dromey classified these quality-carrying properties into four classes: correctness properties, internal properties, contextual properties and descriptive properties. Based on this understanding of quality Dromey presented separate quality models for software requirements specification, design and implementation of software. Each model has its own set of system (product) components, tangible component properties and related quality attributes that are determined by the properties. Attributes are hierarchically ordered into attributes and sub-attributes. A set of high-level quality attributes should ideally be non-overlapping and describe the priority needs for the software. TABLE 5 shows the quality attributes for implementation. Dromey used ISO 9126 (1991 version) as a basis and added attributes "process-mature" and "reusability". The former refers to a mature and well-defined software development process.

TABLE 5 Dromey's quality attributes for software implementation (adapted from Dromey 1996, 37)

FUNCTIONALITY	**RELIABILITY**	**EFFICIENCY**	**USABILITY**
Suitability	Maturity	Time behavior	Understandability
Accuracy	Fault-tolerance	Resource behavior	Learnability
Interoperability	Recoverability		Operability
Compliance			
Security			
REUSABILITY	**PORTABILITY**	**PROCESS-MATURE**	**MAINTAINABILITY**
Machine-independent	Adaptability	Client-oriented	Analyzability
Separable	Installability	Well-defined	Changeability
Configurable	Conformance	Assured	Stability
	Replaceability	Effective	Testability

FIGURE 6 shows how an implementation component (an expression written in a programming language), its quality-carrying properties and external quality attributes are related.

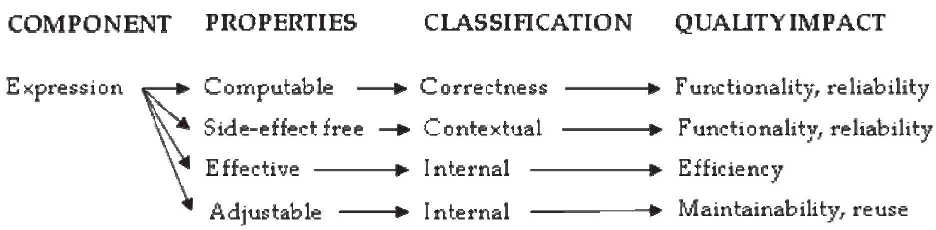


FIGURE 6 Quality-carrying properties of a programming language expression and their effect on quality attributes (adapted from Dromey 1996, 37)

Dromey argued that, if developers are clear about how tangible internal characteristics affect external quality attributes, it is much easier to tune the development process accordingly.

2.2.7 Unified software development process and quality

"The Unified Software Development Process" written by Jacobson et al. (1999) is regarded as a landmark of the popular object-oriented development approach. It presents a method that supports the whole software development life cycle and uses unified modelling language (UML) as a practical tool. The unified process is labelled as "use-case driven, architecture-centric, iterative and incremental". It underlines the role of different actors and stakeholders, and that systems must add value to their users. This turns strongly attention to the system environment and to the relationship between the system and context. The book does not deal with quality explicitly, but the general approach to this aspect can be derived from the text.

Jacobson et al. (1999) divide requirements into functional and non-functional (TABLE 6). Functional requirements specify actions that the system must perform. They specify the input-output behavior of the system, i.e. what the system must do for each user. The authors underline the importance of distinguishing 'key' or 'core' system functions from others. Alternatively, the words "critical" and "crucial" are used. The combining term is "priority". The primary means for capturing functional requirements are use cases. (Jacobson et al. 1999, 37, 114, 445) Nonfunctional requirements, in turn, cover environmental and implementation constraints, platform dependencies, and properties like 'performance', 'reliability', 'maintainability', 'extensibility', 'availability', 'accuracy' and 'security'. (Jacobson et al. 1999, 42, 447) A sub-group of nonfunctional requirements are 'supplementary requirements', i.e. those requirements that are common for many or all use cases. (Jacobson et al. 1999, 117,131)

Discussion of architecture touches on system software, legacy systems and standards. This implies issues of compatibility and integration. 'Usability', 'recovery time' and 'memory usage' are also added to the list of nonfunctional requirements. (Jacobson et al. 1999, 65-68) Further, two other quality factors are mentioned: 'cost' (Jacobson et al. 1999, 114) and 'ease of learning' (Jacobson et al. 1999, 129). TABLE 6 shows the quality model implicit in the unified development process.

TABLE 6 "Quality model" implicit in Jacobson et al. 1999 (diagram is created by the author of thesis)

FUNCTIONAL REQUIREMENTS	**NON-FUNCTIONAL REQUIRMENTS**
Actions that system must perform (usefulness)	Performance Reliability Maintainability Extensibility Availability Accuracy Security Usability Recovery time Memory usage Ease of learning Cost

Jacobson et al. (1999) give very few definitions of quality requirements. One of them is the definition of reliability as "the ability of a system to behave correctly in its actual execution environment" (Jacobson et al. 1999, 448). It is measured "in terms of system availability, accuracy, mean time between failures, defects per 1000 lines of code, and defects per class" (Jacobson et al. 1999, 448). Performance refers to speed, throughput, response time and memory usage (Jacobson et al. 1999, 116).

Nonfunctional requirements are connected either to relevant use cases or classes in models, or managed as a separate list, if they are too generic to be related to individual model objects. UML offers an extension mechanism called "tagged values" for use. (Ahmed & Umrysh 2002, 33, 34)

2.2.8 ISO 9126 product quality model

The basic idea of standards is to define a level of quality that is thought to be acceptable. The world's largest developer of standards is International Organization for Standardization (ISO). ISO is a non-governmental organization and its standards are voluntary, although in some countries taken partly into regulatory use. ISO 9000 family of quality standards, for example, have become a market requirement. There exist several standards concerning software and system engineering. They constitute a rich source of material for immersing deeper into the questions of quality.

ISO 90003 (2004) applies quality standards to the acquisition, supply, development, operation and maintenance of software. ISO 90003 (2004) is written on a very general level, is process oriented and comparable to a checklist. It lists a number of customer related requirements: functionality, reliability, usability, efficiency, maintainability, portability, security and safety. In addition to the requirements listed above, ISO 90003 (2004) notes compatibility or integration by referring to the interfaces between a software product and other products or systems. It states further that some of these characteristics may be critical and all of them should be traceable throughout the development life cycle. Some additional general software properties are mentioned sporadically.

The actual standard of software product quality is ISO 9126, published in 2001. The main part of the ISO 9126 quality documents is devoted to detailed metrics (ISO 9126-2, 3 and 4). It explains also how the quality model and metrics can be used as part of software development life cycle. It indicates which measure is possible in a particular phase, which deliverables can be measured and which metrics can be applied.

A two-part quality model is presented in the first part of the standard (9126-1): a) internal quality and external quality and b) quality in use. The internal and external characteristics are in fact the same, consisting of six main characteristics, which are further divided into measurable sub-characteristics. The difference between the three quality layers lies more or less in the owner of the quality. Internal quality is the quality of the intermediate deliverables of system development process. External quality and quality in use refer to the quality of the final product. Quality in use consists of four characteristics: effectiveness, productivity, safety and satisfaction. It is the combined effect of the main characteristics and the four characteristics that describe the ability of the product to help users to achieve specified goals with effectiveness, productivity, safety and satisfaction in a specific context of use. FIGURE 7 lists the characteristics of the two parts of ISO 9126 quality model: 1) internal and external quality, 2) quality in use (upper part above the horizontal brace).

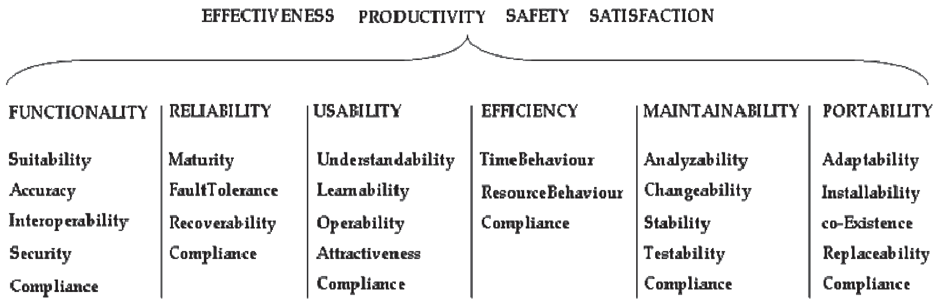


FIGURE 7 ISO 9126: Internal and external quality characteristics together with quality in use (adapted from ISO 9126-1 (2001), 7, 12)

All in all, ISO 9126 documents give lots of space to the differentiation between "internal" and "external" qualities (characteristics) and views. The latter is associated with user and the use of the product during testing or operation in an intended system environment, and the former with the developer and non-executable intermediate products (requirements definition, design specifications, source code, etc.). Appropriate internal qualities are a pre-requisite for achieving certain external qualities. In addition to the notions of 'characteristic' and 'subcharacteristic', the standard exploits the concept of 'attribute'. It is a measurable physical or abstract property of an entity that can also be seen from internal or external viewpoint. These three constructs have a hierarchical relationship: attributes as smallest elements influence the subcharacteristics that are grouped into characteristics (accordingly subcharacteristics influence the characteristics). Internal attributes are the most measurable parts of the model.

ISO 9126 gives each characteristic and subcharacteristic a definition. 'Effectiveness', a quality in use characteristic, is defined as "the capability of the software product to enable users to achieve specified goals with accuracy and completeness in specified context of use" (ISO 9126-1 2001, 12). 'Functionality', in turn, has the following definition: "the capability of the software product to provide functions which meet stated and implied needs when the software is used under specific conditions" (ISO 9126-1 2001, 7). And the subcharacteristic 'accuracy' is defined as "the capability of the software product to provide right or agreed results or effects with the needed degree of precision" (ISO 9126-1 2001, 8). The common subcharacteristic 'compliance' refers to adherence to standards, conventions and regulations with respect to the characteristic in question. Internal functionality metrics, for its part, is designed to predict if the software product will satisfy prescribed functional requirements. Computational accuracy, for example, can be predicted by counting the number of functions that have implemented the accuracy requirements and comparing it to the total number of functions with accuracy requirements. External functionality metrics, in turn, measures the functional behaviour of a system containing the software product. From that viewpoint computational accuracy can be measured by recording the number of inaccurate computations per operation time. Finally, effectiveness can be measured for example by comparing the total number of tasks attempted with the number of tasks completed. The quality in use metrics part of ISO 9126 notes the context of use as important element of quality modelling. Context of use is determined by the user, task, and physical and social environmental factors. In connection with performing product evaluation users, their goals and the environment must be identified.

ISO 9126 is also aware of certain relativity in quality modelling. It notes that users do not always know their real needs or the needs may change. Different users can have different operating environments. In addition it is impossible to consult all possible types of users. Therefore the goal is not necessarily to perfect quality, but sufficient quality for each specific context.

2.2.9 Architecture analysis and quality attributes

One of the theoretical backgrounds for the study was architecture analysis as presented by Svahnberg, Wohlin, Lundberg & Mattsson (2003) and Svahnberg & Wohlin (2005). It looks for methods to help in deciding which quality attributes support which software architectures and vice versa. This is needed because of the different views of stakeholders on quality requirements and because of the differences in how the developers understand weaknesses and strengths of software architectures. During the analysis each actor creates a list of prioritized quality attributes and compares architecture candidates for each attribute and, vice versa, attributes for each architecture. The method recommends grouping attributes into categories to facilitate their prioritization. The overlap between categories should be minimal. The different results are then combined and used to calculate which architecture best meets the quality requirements. In a real context the architecture analysis takes as input a target system, a number of concrete architectures, and a relevant set of quality attributes. The relevance of attributes means significance to the domain and business model in question.

In the study Svahnberg & Wohlin (2005) used five different architecture types: layered, pipes and filters, blackboard, model-view-controller and microkernel. Quality attributes were taken from ISO 9126 (2001) and categorized according to this standard: functionality, reliability, usability, efficiency, maintainability and portability. In the Svahnberg et al. (2003) the attributes were: cost, functional flexibility, performance flexibility and security flexibility. The authors note that attributes in categories interact, support or compete with each other. As a decision support method, when comparing and choosing between quality attributes and architectures, Svahnberg & Wohlin (2005) use Analytic Hierarchy Process (AHP) (Saaty 1980). The study shows that usability, reliability, maintainability and portability support each other, whereas functionality and efficiency are difficult to combine with them. Further it shows, for example, that pipes and filters architecture supports best efficiency, model-view-controller usability and layered maintainability. It is important, however, to keep in mind that the results reflect at first hand participants' opinions and only indirectly factual relationships between architectures and attributes.

2.3 A summary of theoretical background

The term "model" is used in the study in the sense of a blueprint for something to be realized (see discussion in section 2.1 above). Accordingly an information system quality model is a blueprint for the constituents of quality of an information system as a technical artefact being developed and operational in the context of human societies, organizations or personal lives. Models can possess different levels of generality. In a three layered view there are at the bottom quality models for individual real information systems, above them general models for all information systems or general models for certain kind of information systems, and on the top a model for all information system quality models called meta-model. The latter, the subject matter of this study, can also be called a theory of information system quality models.

As was stated in the introductory chapter, one does not often encounter discussion about the ontology, i.e. meta-level, of holistic quality models. What are usually presented are general sets of quality attributes accompanied with instructions on how to measure the attributes. In addition, individual qualities like usability, security, reliability, etc. have got a lot of attention. Due to this scarcity of theory about quality models the theoretical background for the study has been formed on basis of a limited number of sources: five well known general quality models completed with a single attribute (usability) model, a handbook representing the prevailing object oriented development paradigm, one system architecture evaluation method (architecture analysis), relying on knowledge about quality attributes, and with software business peoples view of quality. Comparison to the few other meta-level initiatives is done in chapter 7 that presents the final version of meta-model. The advantage of this approach is also having a relatively "clean table" to start with meta-level modelling (cf. discussion about criteria for grounded theory in section 3.2). ISO 9126 (2001) version is a kind of watershed between the theoretical background and the newer models. The latter are discussed in section 7.5 and occasionally referred to in connection with presentation of the final meta-model. Limitations of background theory are discussed at the beginning of next chapter. Valid constructs that were found are, in turn, referenced and a summary of them given in connection with initial quality meta-model in chapter 4.

3 RESEARCH QUESTIONS AND DESIGN

The previous chapter laid the groundwork for the study by discussing models and modeling and presenting the literature that is used as background for building the initial quality meta-model. This chapter discusses first the limitations in background theories. Then it explains how the study tries to address these limitations. Next the chapter presents the overall research design as well as detailed research questions and in the last section discusses the case studies from methodological viewpoint.

3.1 Limitations in background theory

The literature review in section 2.2 covered five well known traditional quality models (McCall et al (1977), Boehm et al. (1978), Grady & Caswell (1987), Dromey (1996) and ISO 9126-1 (2001)). To gain a wider perspective on information system quality, the theoretical background was completed with a collection of writings from software business executives (Kearns & Falls 2004), a handbook of object oriented software development (Jacobson et al. (1999)), a commonly recognized model of one single quality aspect, namely usability (Nielsen (1993)) and finally architecture analysis, a special approach to quality modelling as represented by Svahnberg et al. (2003) and Svahnberg & Wohlin (2005). The few existing newer theories about quality modelling are in this study seen as reactions to traditional models and as alternatives or competitors to the proposed conceptual framework and will be therefore discussed later in section 7.5 where the final version of meta-model is evaluated.

Traditional quality models have been criticized for several reasons, mainly because not bringing orderly into attention all important aspects of information system quality or quality modelling. This holds true even more with theories and methods whose focus is not on quality issues, like unified software development method, or which focus only on certain aspects of quality, like

Nielsen's usability model or architecture analysis. Some individual examples of critics are given below:

Older quality models:

- McCall et al. (1977) do not consider directly the functionality of the software product. (Ortega et al. 2003)
- Grady and Caswell (1987) fail to take account of the software product's portability. (Ortega et al. 2003)

ISO 9126:

- There are no guidelines on how to provide an overall assessment of quality. (Côté et al. 2007, Chua & Dyson 2004)
- There are ambiguities in the way ISO/IEC 9126 model is structured in terms of characteristics and sub-characteristics. (Jung, Kim & Chung 2004, Kitchenham & Pfleeger 1996)
- On one hand some sub-characteristics (e.g. usability) should be split into more specific ones, on the other hand some of them (e.g. understandability and learnability) should be merged. (Chua & Dyson 2004)
- ISO 9126 focuses on developers' view of quality at the expense of evaluating the quality from the user's point of view. (Côté et al. 2005)
- Siakas & Georgiadou (2005) saw it necessary to complete ISO 9126 main attributes with two new ones, extensibility and security. Another example of needs to modify ISO 9126 is Villalba, Fernández-Sanz, Cuadrado-Gallego & Martínez (2010).

In addition to the particular references above, following general observations can be made regarding the theoretical background presented in previous chapter:

- The concept of quality is left undefined or there is a notable tendency to equal quality to internal characteristics of a product (e.g. in ISO 9126).
- The difference between quality requirements and other requirements is not clearly explained.
- Traditional quality models present a fixed and limited set of attributes and attribute categories. It is, however, impossible to build a complete list of quality attributes. In practice new areas of concern emerge continuously and different applications and environments put different attributes into focus.
- Large general quality models, especially with extensive metrics, do not meet post-methodology era's agility requirements. In addition, many quality attributes are by nature relative, fuzzy, refer to complex relationships, and consequently not easy to define and measure.
- Traditional quality models don't give guidelines on modeling individual quality attributes, except with respect to metrics and general definition.

And accordingly, none of the background theories gives guidelines for modeling together both the overall quality and individual attributes.
- The meta-level, the ontology behind models is seldom discussed. For example, what are the elements of a proper overall quality model or an individual attribute model.

3.2 Overall research design and questions

The study assumes that the problems discussed in the previous section can be overcome on a higher level of abstraction by creating a consolidated conceptual framework or meta-model, which is used as a flexible template for all lower level quality models. The term "model" is here a synonym for theory (cf. discussion about the terminology in section 2.1 above), namely theory of quality models. Accordingly, the main goals of the study are 1) to define the concept of information system quality, applicable to information systems as technical artefacts in their context of development and use, so that quality requirements can be separated from requirements in general and 2) to develop a blueprint for a quality meta-model applicable as well to overall quality as to individual qualities, and covering all essential elements needed in creating system specific quality models. The ideal of comprehensiveness, i.e. covering all essential aspects, differs from the more limited model concepts or sub-models presented in some writings like Siakas & Georgiadou (2005) that define a quality model simply as "a set of characteristics and the relationships between them". The main research question concerns the creation and validity of the proposed conceptual framework.

The overall research approach can be characterized as mixed. It is a combination of conceptual analysis, theory-evaluating and theory-creating research (approaches are presented in detail by Järvinen (2001)). On the one hand the proposed information system quality meta-model is derived from existing quality models and other selected literature by analysing, combining and generalizing from them (Järvinen 2001, 31). On the other hand it is shaped from empirical observations in quality modelling case studies. The approach can also be characterized as qualitative. This holds true for the produced and gathered data as well as for the analysis of it. The only quantitative method used was analytic hierarchy process (AHP) as presented for example by Svahnberg et al. (2003).

A theory or model includes a definition of the area (boundary) where it is applicable, key constructs and the values they can take and the relationships between constructs. Variables associated with constructs define the states in which the constructs can be. (Dubin (1969), Kaplan (1964) and Weick (1984) according to Järvinen (2001, 18)). Following these lines of thought, the quality meta-model created in this study is applicable to information systems as end or intermediate products of development activity. Designing the development process itself, in turn, requires a separate, process quality meta-model. This important dichotomy is noted for example by Hevner, March, Park & Ram (2004, 78) in their article about design science research. Key constructs of quality meta-model, their values and relationships are discussed in connection with initial and final meta-model presentations. Gregor (2006), like Järvinen (2001) above, lists also components common to all theories. In addition to boundaries (scope), constructs and construct relationships, she notes means of representation. In this respect the quality meta-model is described in words, diagrams and prototypes.

Gregor (2006) classifies theories in information systems based on the primary goals of theory and lists the following goals: analysis and description, explanation, prediction and prescription. The theory types, in turn, are (adapted from Gregor 2006, 620):

- Analysis: theory does not extend beyond analysis and description.
- Explanation: in addition to analysis theory provides explanations but does not have testable propositions and does not aim to predict.
- Prediction: in addition to analysis theory provides predictions and has testable propositions but no causal explanations.
- Explanation and prediction: in addition to analysis theory provides predictions, testable propositions and causal explanations.
- Design and action: theory gives prescriptions for construction an artifact.

In the light of above typology the quality meta-model belongs to the first and last categories: analysis, design and action. It has also some flavour of explanation through case studies and by viewing, for example, prioritization of quality attributes in relation to actor perspectives. As a theory the meta-model does not assert anything testable about the relationships between its elements. It cannot be used to create natural science like predictions. On the other hand the model postulates a set of concepts, their logical relationships and prescribes how quality models ought to be structured (cf. discussion about constructive research in Järvinen 2001, 88-89). The following proposition, for example, that could be derived from the final meta-model has inevitably some prescriptive or practical flavour:

An individual information system quality characteristic is properly described, if (1) it is named, given a general definition and assigned to a domain, (2) attributed to the information system as a whole or to some of its constituents, (3) prioritized and related to other attributes in the set, (4) further defined through use cases, scenarios, indicators, contributors, and an abstract model representing the respective system-context relationship and (5) a relevant measurement arrangement for finding out the actual attribute value has been designed.

Due to its nature as a theory for analysis, explanation, design and action, the quality meta-model cannot be tested in a strict sense. This is why the second aspect of the research approach was above called theory-evaluating rather than theory-testing. Nevertheless it must be evaluated in some way. Usually an assessment is based on comparison to previous theories, i.e. to theoretical background. Frameworks like ISO 9126 are, however, basically just general attribute lists and quality metrics specifications for all information systems combined with some dispersed theoretical thoughts about quality and quality modelling. Accordingly in the chosen theoretical background there are no actual higher level theories to be compared with the new meta-model. Means left for evaluation are therefore in the first instance 1) implications for action and 2) ideas for new artefacts. (Järvinen 2001, 32-34) The former is suggested by Marcus (1983), the latter is the viewpoint of constructive studies. Evaluating meta-model this way means that the system specific quality models are interpreted as artefacts built by applying the meta-model. In addition, the main steps of modelling process, corresponding to the model elements, can be seen as implied by the meta-model. The instantiated quality models are like "deduced" from the theory and the prediction is that they are useful. The study must then to evaluate how useful these actions and artefacts were. There are also two other ways to ease the strict, natural science type, theory testing requirements. First, even if the meta-model as a theory does not contain assumptions that can be used to create hypotheses, it has a set of constructs that represent items in the domain. These constructs can be compared with the real world of the three case studies. Secondly, traditional quality models can, however, be treated as "pseudo theories" about quality models and compared with the meta-model. Because none of the traditional quality models were in practice used parallel with the meta-model the comparison must be done on theoretical level.

Research methodological background for evaluating the theory can be sought, in addition to what was discussed above, from grounded theory. (Järvinen 2001, 65 - 67) The first criterion for grounded theory, namely fitting to the substantive area, coincides with the requirement that constructs should represent items in the real world, discussed above. Indirect argumentation can also be used by observing how the system specific models fit to the reality of their contexts including system development activities. The remaining three criteria of grounded theory are: 1) the theory should be comprehensible and make sense to the participants, 2) the theory should be general enough to include sufficient variation and 3) the theory should provide control with regard to action. With respect to the first criterion the study again assumes that comprehensibility of system specific models predicts the comprehensibility of the meta-model.

In a recent article Urquhart, Lehmann & Myers (2010) discuss grounded theory in information systems studies as well, but list a different set of distinctive characteristics for grounded theory. 1) Main purpose is theory building. 2) In general no preconceived theoretical ideas. 3) Comparing collected data constantly to existing constructs. 4) On the other hand, constructs established so far direct collection of further data. This is called theoretical sampling. All this complies with the design of the study at hand. The main goal is clearly theory (meta-model) building. The preconceived ideas are actually very few. The theory is created by generalizing from existing quality models. And, as Urquhart et al. (2010) state, if the researcher starts from existing theory, the aim of grounded theory is to improve the theory. Finally, the interplay between existing constructs and collected data is exactly what happens along the three case studies leading to a final version of the theory. The concepts in the initial meta-model are comparable to "seed concepts" in Urquhart et al. (2010, 362). The analysis of constructs into properties and discovery of relationships between them, however, is not yet as rigorous as it could be. Urquhart et al. (2010) discuss also the three levels of theory in grounded theory method. These levels, in turn, are comparable to the three levels of abstraction in quality modelling.

Hevner et al. (2004) take a closer look at design science in information systems research. Information technological artefacts can be constructs, models and instantiations (implemented and prototype systems). The quality meta-model belongs to the two first categories. Hevner et al. (2004) give also guidelines for research and evaluation of the artefact. Fundamental principle is creation of a purposeful artefact that yields utility for a specified problem. Information technology artefacts can be evaluated in terms of functionality, completeness, consistency, accuracy, performance, reliability, usability, fit with the organization and other relevant attributes (Hevner et al. 2004, 85). With respect to quality meta-model the specific problem is creation of lower level quality models, which in turn are blueprints for a quality product. Accordingly the attributes relevant to evaluation are completeness, consistency and usability. Finally, design science must provide clear contributions of one or more of the following: novelty, generality and significance. Contributions of the study at hand are discussed in the introductory chapter and sections 7.5 and 8.2.

Based on the above considerations the more detailed research questions helping to test the validity of the meta-model can be formulated and grouped as listed below.

Usefulness of new artefacts and actions:

> Q1 Can the system specific quality models and the modelling process be considered as useful and satisfactory?
> Q1a How useful is the process of implementing the meta-model and how useful are the resulting system specific models?
> Q1b Which inadequacies are found in the process or system specific quality models and why?
> Q1c Does the meta-model assist in controlling information system quality?
> Q2 Does the use of meta-model cause bearable amount of overhead in terms of time and other resources?

All the above questions could be asked about traditional quality models as well. This is, however, outside the scope of this study. Traditional quality models were used only to a very small extent as part of the third case study.

Conformity with and suitability to the real world:

Q3 How well do system specific model elements represent the real world items and diverse data in the case study contexts?

Comprehensibility to the participants:

Q4 Is the meta-model comprehensible and does it make sense to actor-informants?

Comprehensiveness, generality and flexibility of the model:

Q5 Is the meta-model comprehensive but at the same time distinctive enough?
Q5a Does the meta-model cover all essential aspects of information system quality?
Q5b Does the meta-model guide in modelling overall quality as well as individual quality attributes?
Q5c Can quality attributes be differentiated from requirements in general using the conceptual framework?
Q6 Is the meta-model general enough to be applicable to a variety of contexts?
Q7 Does the use of meta-model provide flexibility in quality modelling without loosing essential aspects of quality out of sight?

Comparison to theoretical background:

Q8 Does the meta-model describe and explain the information system quality and quality models more comprehensively and sensitively than background theories? (Järvinen 2001, 32)

Quality modelling in practice as part of overall system development:

In addition to the creation of a conceptual framework for product quality modelling, the study aims in second place at finding clues how to use the meta-model in practice. From this viewpoint the extra detailed research questions are as follows.

Q9 What are the implications of case studies for the process of using the meta-model in connection with the development process?
Q9a What is the most effective order of instantiating model elements?
Q9b Which tools and arrangements can help in applying the meta-model?
Q9c How does quality modelling relate to other main development activities?

After 1) initial literature review and identification of limitations in background theories the study proceeded as follows. 2) Relevant concepts found in background theories were generalized, completed with some additional ones and combined to form an 3) *initial information system quality meta-model*. This meta-model was conceived as a model both for general and system specific quality models. 4) Parallel to this detailed research questions (presented above) were formulated concerning the validation and use of meta-model. Then 5) two different quality modelling case studies were carried out where the meta-model or parts of it were used to produce prototypes of system specific quality models for real information systems in order to evaluate the conceptual framework, i.e. to confirm or falsify the meta-model and refine it. The aim was in first place to validate the overall structure of meta-model, not the details of its elements. After 6) analysis of case study results an 7) intermediate version of meta-model was created. A 8) third case study was used to evaluate the intermediate meta-model and 9) after analysis of results it was adjusted to be 10) the *final meta-model*. At the same time the model was made more rigorous by developing accurate construct definitions and assumptions. All system specific models created during case studies can be characterized as prototypes, not as complete quality models for the systems in question. 11) Further literature review was carried out parallel to analysing the results of all three case studies and developing the final version of meta-model. FIGURE 8 depicts the overall research process. The general research design that occurred along the first steps is positioned on top of the diagram.

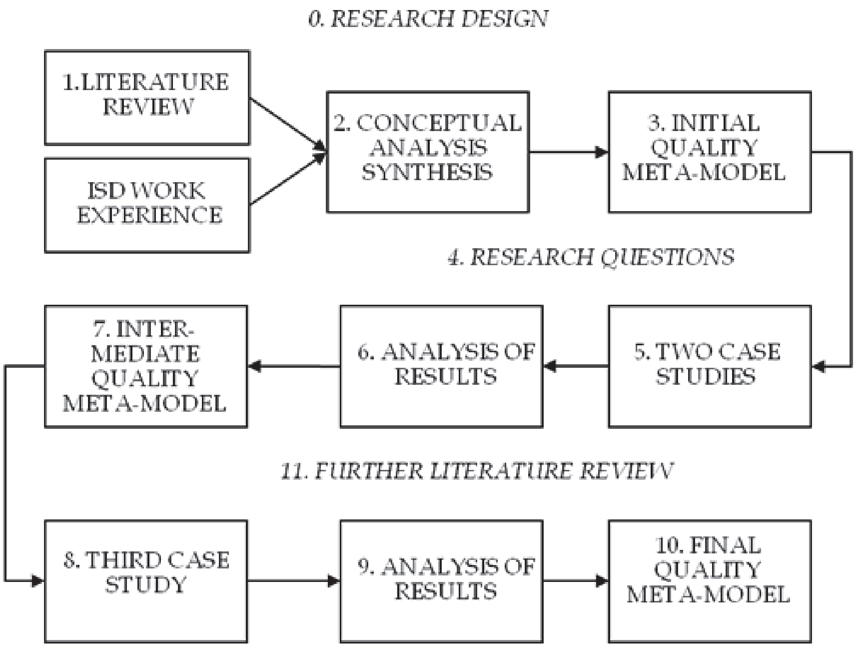


FIGURE 8 Overall research process

The research process can also be described with respect to the levels of abstraction in quality modeling (see section 2.2 above). FIGURE 9 depicts this view. Traditional quality models that belong to the general level were 1) generalized further to create the initial quality meta-model. The meta-model was then used as a template in order 2) to create system specific instance level models in the case studies. Traditional quality models were used in third case study 3) to assist in identifying quality attributes. Finally, 4) research findings were in turn generalized in order to refine the meta-model. Dashed lines indicate a possibility of using system type specific or other general quality models in the process.

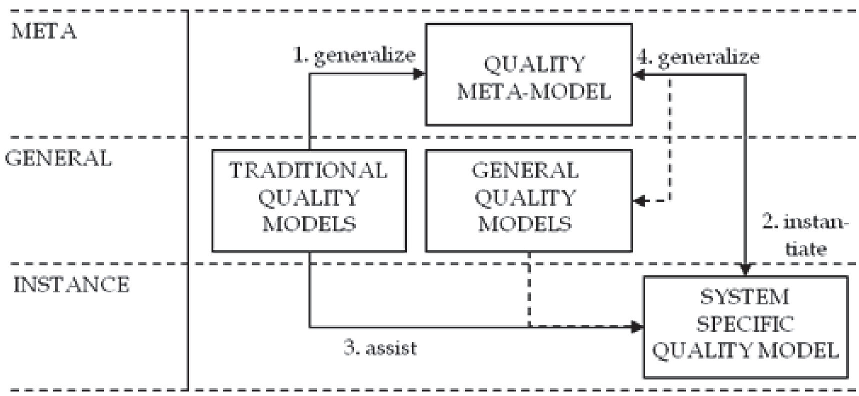


FIGURE 9 Research process and levels of abstraction in modelling

As a first limitation of the study it must be noted that despite the use of three different case studies the observations are quite singular. It relates to two general problems of case study method, namely replicability and generalizability (Lee 1989, 35). Further, Benbasat, Goldstein & Mead (1987) note two sources of lack of objectivity: researcher's stake in achieving a successful outcome and difficulties when used techniques are applied in other situations by people less knowledgeable that the researcher. In addition, due to limited time frames with respect to African context and lack of financial resources to expedite the process, the three instantiated quality models were not perfect at the time of closing cases. The author is well aware of these problems implied in the approach taken. The burden of solving them is upon repeated application of the model in practice. Finally, theories are often what can be called an approximation and not full-blown theories. (Weick (1984) according to Järvinen (2001)) This holds also true for the study at hand. Its focus is on a framework with interrelated concepts that should be taken into account in quality modeling. It does not go to the level of deriving a system of testable descriptive propositions from the assumptions and the degree of conceptualization can be described as "intermediate" (cf. Urquhart et al. (2010, 365-368)).

3.3 The case studies – methodological viewpoints

According to Yin (2009, 7) any research method can be used for exploratory, descriptive or explanatory purposes. Case study is, however, particularly appropriate research method when theory is in an early stage and experiences of actors are important. Knowledge of practitioners is used to develop theory from it (Benbasat et al. 1987). Järvinen (2001, 62) also characterizes this kind of study as exploratory. Yin (2009, 13) sets three criteria for using case study: 1) a "how" or "why" question is being asked 2) about contemporary set of events and 3) the researcher has little or no control. The development of a quality meta-model for information systems meets well the above criteria. As was stated in introductory chapter and previous section, only a handful of theories about quality modelling on meta-level have been published so far. Secondly, views on quality are, as will be seen in following chapters, to a substantial extent related to actors' background and experiences. Next, the study asks among others the question "how should one build a quality model". Finally, the phenomena under study are contemporary and the researcher tried, excluding the format of quality models, to control the process as little as possible.

In addition to the above mentioned principles there is no exact standard definition for case study. Therefore Benbasat et al. (1987) stake out three categories: application descriptions, action research and actual case study. The latter is characterized by researchers being observers rather than participants. In the first category the author of description does not conduct research. In action research the researcher becomes a participant but carries out research at the same time. With respect to this categorization the three quality modelling case studies described in chapter 5 have a flavour of action research. The researcher was more or less (at least in the first case) participant in implementing the information system and wanted at the same time to evaluate a modelling technique. This simultaneous observation and participation, and forming a team that includes the researcher and other subjects as co-participants is also noted as a key characteristic of action research by Baskerville (1999). Further major characteristics of information systems action research according to Baskerville (1999, 6-7) are:

- Research aims at increased understanding of an immediate social situation.
- Research simultaneously assists in practical problem solving and expands scientific knowledge.
- Action is performed collaboratively and enhances the competencies of the actors.
- Research is primarily applicable for the understanding of change processes.

The immediate social situation in the three case studies is the introduction of a new information system. The practical problem is how to create a system that fulfils the expectations of stakeholders. And the studies obviously increase the actors' competencies with respect to information systems. Only the change process was not actually in the focus of study. Accordingly, the case studies did not start with a specific diagnosing phase (cf. Baskerville 1999, 13-15). Even other typical phases of action research – action planning, action taking, evaluating and specifying learning – were only implicit. The cyclical iterative character of action research, in turn, is realized through three different cases. Being iterative is also characteristic of design science (Hevner et al. 2004, 88,89). Baskerville (1999, 12) also notes that an ideal domain for action research is new or changed system development methodologies. Quality application of quality meta-model is such a domain.

Actual case studies conform to following key characteristics according to Benbasat et al. (1987, 371):

- Phenomenon is examined in a natural setting.
- Data are collected by multiple means.
- One or few entities (system specific quality model) are examined.
- The complexity of the unit is studied extensively.
- No experimental control or manipulation is involved.
- The investigator does not specify independent and dependent variables in advance.

Benbasat et al. (1987) discuss also further aspects of case study like unit of analysis, single vs. multiple cases, site selection, data collection and analysis.

Unit of analysis

In the three case studies the primary unit of analysis is the system specific quality model, including the concept of information system quality, and the secondary unit the process of using a meta-model as a template for the former. From organizational point of view the first case deals mainly with the statistical section of ministry's policy and planning unit, the second case with a country office of an international aid organization and the third case with different departments and sections of a ministry plus development co-operation project's program office.

Single-case vs. multiple-case

According to Benbasat et al. (1987) multiple-case design is suitable to theory building or theory testing research like the TANZANIT-study at hand. Multiple-case design provides also the opportunity to cross-case analysis and forms a broader basis for generalization.

Procedure

The studies followed a general case study procedure (Järvinen 2001, 67-73):

1) Selecting the case
2) Selecting the data collection techniques
3) Entering the field
4) Analysing data
5) Shaping theory
6) Comparing to literature
7) Closing the case

Research activities in the field were harmonized with the context, i.e. with all other development activities of the information systems in question.

Site (case) selection

All three cases were selected on practical grounds: the opportunity to get involved emerged through personal contacts or work assignments. They don't represent any unique or revelatory cases (Benbasat et al. 1987, 380), unless location in East-Africa is regarded as such. They provide, however, examples of different information system categories. The first case study was carried out in connection with Education Management Information System (EMIS) development in Tanzania, the second in connection with website development in Mozambique, and the third in connection with land registration system development in Zanzibar. This provides for possible contradictory or otherwise different results.

Approaching the first site, Ministry of Educations and Culture in Tanzania, was most demanding. The correspondence with the ministry started two years before entering the field. Finally in 2004, after getting recommendation from the University of Dar es Salaam, the researcher got a research permit (APPENDIX 1) from Tanzania Commission for Science and Technology (COSTECH). The permit was later renewed for 2005 and 2006. The approach to the other two sites was easier. Both cases appeared through researcher's work as an ICT consultant for the organizations in question. In none of the cases did the host organization or employees get any direct financial compensation for their contribution to this study. This caused some difficulties in getting actors involved in a culture where personal benefits are usually expected. The benefits for the organizations or the individuals were mainly indirect, such as learning, insights and expected quality of the systems under development. Most of the participants in all three cases had no previous experience in information system development or quality modelling.

Data collection methods

Data was gathered from several sources: documents, interviews, direct observation and artefacts. Documentation regarding the first case covers system development and activity plans written by the ministry (MoEC), reports written by the researcher for the customers in different occasions, statistical survey forms and booklets produced by the ministry (MoEC), reports and plans written by donor organizations and consultants. In the second and third case study the role of documentation produced by the "customer" organization, donors or other consultants was smaller. Interviews and related techniques included questionnaire, analytic hierarchy process (AHP), unstructured and semi-structured interview, workshop and group work. AHP was originally developed by Saaty (1980) and has been used in connection with information system quality studies e.g. by Svahnberg et al. (2003). The majority of interviewees were selected in advance considering each system and its stakeholders. Additional informants were, however, added according to needs. Direct participant observations have been recorded into handwritten notebooks, digital research plans, reports, power point presentations and excel sheets. Finally, "physical" artefacts include all quality model prototypes (and assisting drawings) created by the researchers together with the actors as well as system diagrams drawn by the local software company in EMIS case. In addition the training material produced by the researches in the course of first and third case studies can be counted as data about the cases. Additional details of data collection methods can be found in chapter 5 according to their use as part of quality modelling process.

Data analysis

Analysis of data was carried out in connection with presentations given to the stakeholders, writing articles for conferences and magazines and writing the thesis itself. Sharpening constructs was a continuous process and drafts of meta-model with related diagrams were altered accordingly. Comparison to literature was completed in connection with drafting the final meta-model. Theoretical saturation was achieved by the time of third case study, when the researcher concluded that incremental learning had become fairly minimal with respect to the goals of the study.

TABLE 7 summarizes the settings of the three case studies. In addition to what was discussed above, it shows that the researcher's role was different in the first case where the researcher did not participate into actual system development. In the two others he was in charge of designing and programming the systems.

TABLE 7 Summary of the three case study settings

	LOCATION	INF SYSTEM	ACTOR ORGANIZATION	RESEARCHER'S ROLE
CASE 1	Tanzania Dar es Salaam	Education Management Information System (EMIS)	Ministry/Statistical Section	Researcher-Consultant
CASE 2	Mozambique Maputo	Website	International Aid Organization/ Country Office	Developer-Researcher
CASE 3	Zanzibar Stone Town	Land Registration System	Ministry/ Departments and Development Consultancy/Project Office	Developer-Researcher

The different settings of case studies allow to some extent for replicability. If the instantiated system specific quality model was useful in first case it should be useful in other cases under different set of conditions. (cf. Lee 1989, 40-41) Similarly, some degree of generalizability can be achieved through these three different empirical circumstances.

4 INITIAL META-MODEL

This chapter presents the initial quality meta-model that was derived from the traditional quality models and other selected literature. The chapter consists of three sections. First one gives an overview of the meta-model and shows it in a diagrammatic form. The second section discusses each model element and its relationship to the theoretical background individually. The last section summarizes main differences between the meta-model and the background theories.

4.1 Overall view

The elements of initial quality meta-model grew up from the analysis of literature discussed above and reflections on author's own experience in software development. The elements are generalizations from objects and phenomena found in the background theory or real life system development. The traditional quality models were taken both as one kind of existing theory and as examples of quality models. This is a legitimate way of creating a theory. It includes both deductive and inductive features. Järvinen (2001,17,27,30-31), for example, takes up building theory on one hand deductively by comparing and combining existing theories, and on the other hand inductively by generalizing from observations or by interpreting old results in a new way.

At the beginning of the study, before entering into the field for the case studies, the initial quality meta-model was described only verbally (see Finne 2005). FIGURE 10 shows a diagrammatic representation of the textual form.

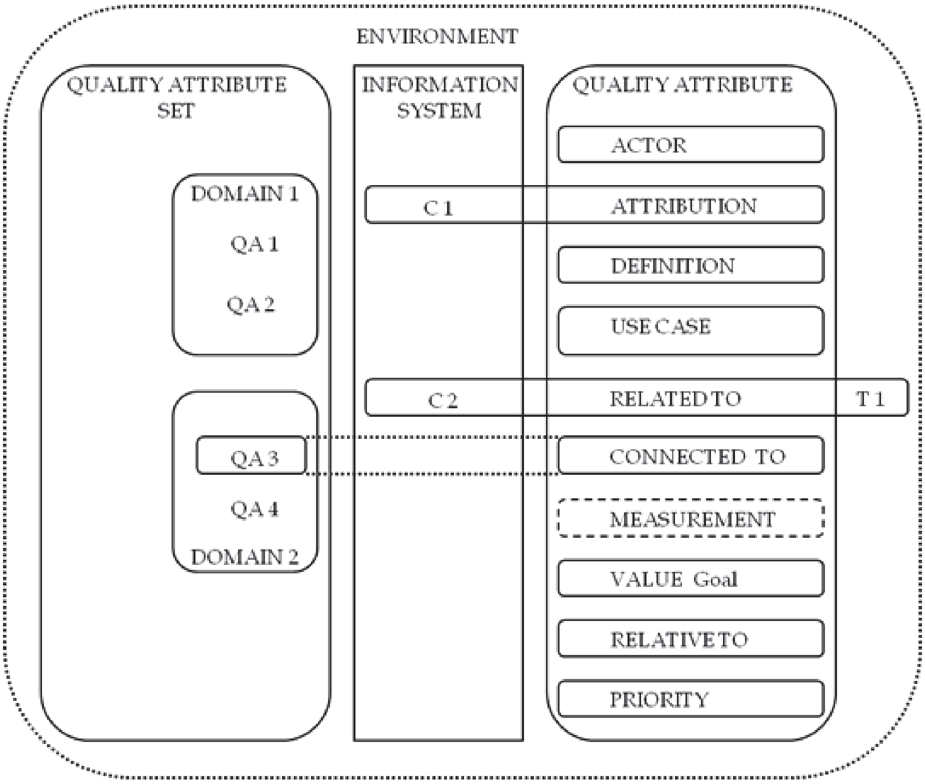


FIGURE 10 Diagrammatic representation of initial quality meta-model

The meta-model has two main parts or sub-models: 1) on the left-hand side *quality attribute set model* with attributes (QA1, etc.) categorized into domains (DOMAIN1, etc.) and 2) on the right-hand side a *quality attribute model*. In the initial meta-model the order of elements does not indicate the order of application or use in practice. The elements of attribute model are constructs needed in modelling individual qualities. Dashed line around measurement element indicates that metrics is not a genuine part of product quality model (see discussion below in connection with the element). Information system with its constituents (C1, etc.) in the middle indicates that both model parts apply to an information system as a whole or to some of its constituents. For the sake of consistency all element names are written in singular form. In a system specific model, however, each element can have multiple realisations. For example there are normally several attribute models, an individual attribute model has many definition elements, etc. The extended rounded rectangles of attribute sub-model in FIGURE 10 symbolize following relationships:

1) ATTRIBUTION: a quality attribute can be attributed as well to the information system as a whole as to some of its constituents (C1 in the figure).

2) RELATED TO: The definition of a quality attribute relates the information system as a whole or some of its constituents (C2 in the figure) to certain things (T1 in the figure) in its environment.
3) CONNECTED TO: A quality attribute can be connected to another quality attribute (QA3 in the figure) in the same attribute set.

All the meta-model elements are discussed separately below in section 4.2. It can be seen from FIGURE 10 that in the initial meta-model much attention was paid to the quality attribute part that has not been fully opened up in traditional quality models that are usually limited to naming, defining generally, categorizing and measuring the values of quality characteristics. In fact, a separate quality attribute meta-model existed - and had a diagrammatic form (FIGURE 11) - before the field work started. In the attribute meta-model diagram *relativity points* (R) indicate elements whose values or features are in one way or another relative. *System* refers to the information system or its constituent that possesses the relevant qualities. *Structure* (inner circle) refers to the relationships between the system's constituents and *behaviour* (outer circle) to their functional behaviour. *Dictionary definition* relates to a general definition of a quality attribute. *Propositions* and *predicates* are more advanced and formal ways to define attributes. *External things* (T) are things in the environment that, according to attribute definitions, have via use cases a relationship to the system. *Measurement* refers to measurement arrangement and process, *value* to the measured value of the quality in question. In the overall diagram above (FIGURE 10) the attribute meta-model is embedded into the information system quality meta-model as a quality attribute sub-model. Dictionary definition, propositions and predicates are included into definition-element. Structure and behaviour, in turn, are hidden into information system element between the two sub-models, and ranking equals with priority. Otherwise the relationship between the obsolete attribute meta-model and the comprehensive quality meta-model can be easily seen. All in all, one can anticipate from both figures that an information system product quality model is a *hybrid model*. And it is not only a blueprint of some internal system features. It is rather a design of an entire state of affairs.

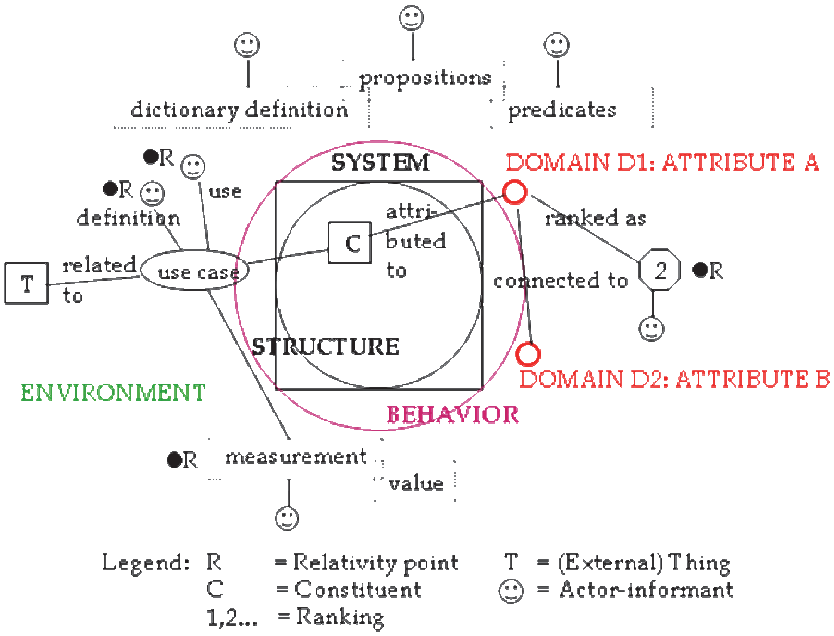


FIGURE 11 Quality attribute meta-model

As was stated in section 2.1 above a meta-model is a kind of template for models. Consequently, the initial information system product quality meta-model is a template for lower level information system product quality models. It tells which things must be specified or modelled with respect to overall quality or individual qualities in order to be aware of them, to understand them and to be able to implement them. The meta-model can also be used to build more general models, for example according to application or system type. The study will recommend an order in which to apply meta-model elements. The detailed format, modelling languages, methods and visualizations used in physical quality models, in contrast, may vary case by case and do not belong to the subjects of the study.

It is evident and clearly recognized that there is a connection between the quality of the end product and the development process. The proposed quality meta-model is, however, intended to apply only to the quality of information system as an end product. The quality of a process would draw into focus a different set of concerns and attributes, like agility, cost-effectiveness, development method and tools, productivity etc. Modeling, implementing and testing (measuring) quality are in this study seen as processes and as integral parts of system development process and system lifecycle. FIGURE 12 (part of FIGURE 1 in section 2.1 above) depicts these different targets (the target of this study highlighted) of quality modeling. The widest scope for a quality model with respect to information systems is the entire system life course process quality model.

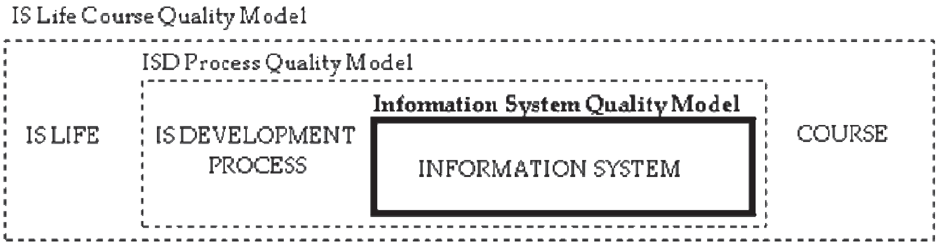


FIGURE 12 Possible targets of quality modelling

While the applicability of the meta-model is restricted to products instead of processes, the framework is not, however, confined as only applicable to a piece of software or to an individual application. It can as well be applied to the quality of an information system as a whole or to the quality of a large integrated system comprised of hardware, software and multiple applications.

4.2 Constructs

This section presents in more detail each element of the initial information system quality meta-model. First, each element is given a non-formal textual definition and then it is related to concepts found in the theoretical background literature. The trace-back relationships are not complete, but sufficient to indicate the connections. Elaborated definitions of quality and quality requirement are not yet given in connection with the initial meta-model. According to the layout of the study these, like the final framework, grow later out of findings and conceptual analysis.

Actor

Actor means a human or non-human actor (e.g. other information system) around the information system under scrutiny. Different types of human actors include people from end users to different stakeholders, who may never use the system, but are somehow affected by it or its products. Actors can be individuals or groups (collective actors). An *informant* with respect to quality modelling is a human actor, who is acceptable for giving some relevant information concerning quality attributes. This information may concern definition, measurement and values, or prioritising of the attribute. (Finne 2005)

In Grady and Caswell (1987) actors dealing with quality were the company's Software Metrics Council and software project managers. Actors were not, however, taken as a separate aspect in the quality model. Nielsen (1993), for his part, was fully aware of different users and user groups and their different views of system and its quality. He advocated even participatory design, where representative users take part in the design phase (Nielsen 1993, 88). Architecture analysis notices clearly that different stakeholders tend to have different views of the importance of various quality requirements (Svahnberg & Wohlin 2005). In ISO 9126 users are part of context of use that must be identified when quality is evaluated. Unified software development process also highlights the role of people (architects, developers, managers, users, customers and other stakeholders) as prime movers in software project. In this tradition the notion of 'user' includes non-human users (other information systems) (Jacobson et al. 1999, 5, 15), from where it was taken to the definition of actor element above.

Information system

It is obvious that one needs at least some understanding about the entity, whose quality is being modelled. In a common view the architecture of an information system is presented as layers, subsystems and components. These things can be called "*constituents*" that are physical or logical (FIGURE 13). Structural relationships of constituents are referred to as "*structure*" (inner circle with solid line in the figure) of information system. *Behaviour* (outer circle in the figure) of system and its constituents is usually viewed separately and contrasted with constituents and structure (e.g. Martin & Odell 1996, 39). *Contents* (inner circle with dashed line in the figure) can also be viewed as a logically independent part of system. This view can be embedded into diagrams depicting the system environment, but as such it ignores the context. It can be called "a simple structural-behavioural view" (Finne 2005). It was used also in the quality attribute meta-model above in chapter 4.1. Constituent and system interfaces are not marked in the figure.

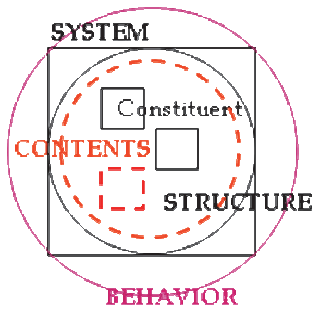


FIGURE 13 Structural-behavioural view of information system

Information system in a form or another is implied in all traditional quality models but its definition has not got much attention. Nielsen (1993), for example, does not present any specific overall view of information system. Svahnberg et al. (2003) and Svahnberg & Wohlin (2005) take a clear architectural view on systems. Dromey (1996) recognizes information system as a set of components. ISO 9126 defines, following ISO 12207 (2008), software product as "the set of computer programs, procedures, and possibly associated documentation and data" (ISO 9126 (2001, 21)). This is similar to the structural-behavioural view in initial meta-model in a sense that it also excludes the context. For Jacobson et al. (1999, 20) a software system is "all the artifacts that it takes to represent it in machine or human readable form to the machines, the workers and the stakeholders". Unified software development process strongly advocates an architectural view of information system (Jacobson et al. 1999, 59-84).

Environment

The term "environment" was frequently used in connection with discussing information system qualities in Finne 2005, but never explicitly defined. It appeared in connection with attributes like 'portability', 'co-existence' and 'impact on environment'. It was also noted that measuring direct or indirect impacts of information system on environment is outside the competence of computer science alone and would mean joint projects with researchers from social sciences. Now environment is included into initial meta-model as a separate element. It refers generally to everything outside the information system that is meaningful with respect to quality modelling.

ISO 9126 (2001) notes environment in the model for quality in use. It is called "context of use". Architecture analysis, in turn, takes up environment indirectly by noting that attribute sets should be relevant to domain and business model in question. The importance of these two models is clearly pointed out by Jacobson et al. (1999) as well. Otherwise little attention is paid to environment in the background literature.

Quality Attribute

A basic English dictionary (e.g. Cambridge Advanced Learner's Dictionary 2008, 1162) defines *qualities* basically as the good characteristics, which something has. A more formal and literate synonym for *property* or *characteristic* is "attribute". With respect to information systems an attribute belongs to the entire system or to some of its constituents. Good characteristics of an information system are closely related to the satisfaction or dissatisfaction of actors around it and they are called "*quality attributes*". A system, without a certain number of quality attributes possessing certain positive values, cannot be regarded as an acceptable information system. This subset of attributes can be called "*vital quality attributes*". (Finne 2005)

Good characteristics of information system have been given different names by authors. McCall et al. (1977) write about "quality factors". Nielsen (1993) uses the word "concern" when referring to usability, acceptability, etc. Further Nielsen uses the expression "usability attribute" for the components of usability like learnability, efficiency of use, memorability, etc. ISO 9126 (2001), like Boehm et al. (1978), uses he terms "characteristic" for different qualities. Jacobson et al. (1999) call qualities "nonfunctional requirements". Architecture analysis, as well as Grady and Caswell (1987)) and Dromey (1996), opts for the term "quality attribute".

Quality Attribute Set and Domain

The problems of coverage, categorization and to some extent relativity are addressed in the proposed quality meta-model using concepts 'collection of quality attributes', 'quality attribute set' and 'domain'. Coverage means here that the quality model covers all important qualities. Relativity, in turn, refers to the fact that actors may disagree on the list of important qualities. In the meta-model *quality attribute set* refers to the quality profile of a particular information system, i.e. to the categorized and prioritized set of important qualities that according to the informants are required from the system in question. The model does not prescribe any quality attributes or attribute categories. In practice, however, a starting point is usually needed. It can be provided with a pool of different quality characteristics that possibly can be attributed to an information system in general or to a certain type of information system. In connection with the initial meta-model this pool is called "palette" or a *collection of quality attributes*. The attributes in a quality attribute set or pool are arranged into groups called "domains". *Domains* refer to major concerns with respect to information systems. In other words, a domain means a field of thought, or thing(s) in connection with which a group of attributes are relevant. It is a solution to the problem of categorizing attributes. Both general collections and system specific attribute sets should be kept dynamic and open to new domains, new attributes and regrouping of attributes.

FIGURE 14 depicts graphically a categorized quality attribute collection that was devised for the case studies. It contains all the characteristics found in ISO 9126 (2001) standard completed with some additional properties taken from the documents reviewed in Finne (2005). The "palette" is divided into sectors according to domains, represented by keywords (attached to small filled circles), that refer to important aspects of information system or its development. General definitions of attributes and domains, not visible in the figure, are essential elements in the palette.

No listing can cover all possible quality characteristics relevant to all possible information systems. Similarly a fixed and non-controversial categorization of quality attributes is probably impossible. ISO 9126 (2001) actually allows that a software product specification can also use its own categorization of quality attributes. Further, the combination and importance of quality attributes varies, for example, according to the type of information system (noted already by McCall et al. (1977)). Nevertheless many attribute listings and groupings have been created and published (see the literature review in chapter 2.2). Nielsen's (1993) model of system acceptability is an example of general, but limited collection of attributes that are grouped into categories. ISO 9126 (2001) and other traditional quality models present larger collections of quality attributes. Main characteristics correspond to Nielsen's categories. Jacobson et al. (1999), on the other hand, put quality attributes under the notion 'nonfunctional requirements' and do not present any sub-categorizations. Architecture analysis, in turn, talks of attribute sets relevant to particular systems and recommends grouping attributes into categories. Practically all quality models contain the idea of a fixed number of attributes that can be brought together, arranged into smaller groups and related to each other in a fixed manner.

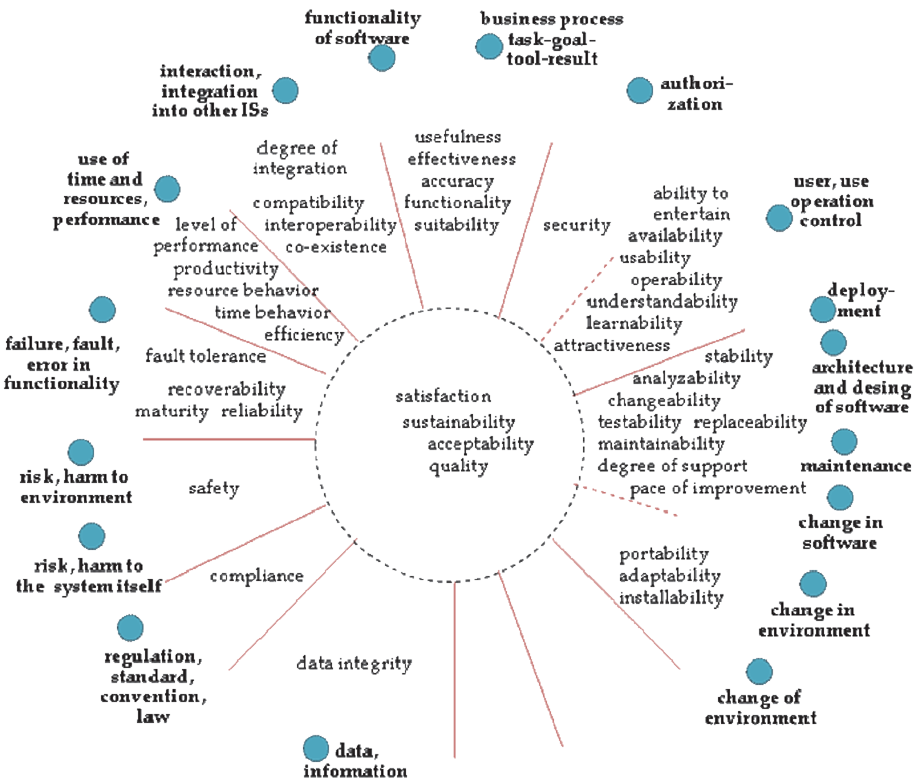


FIGURE 14 A general collection of quality attributes

Priority

Priority refers to the ranking of attributes, i.e. defining their importance in relation to each other. Prioritising attributes can be logically differentiated from prioritising system constituents, structures or behaviour. The qualities of important constituents are obviously more important than those of less important constituents. One can, however, also rank qualities just on the level of entire system without attributing them to a specific constituent. Grady and Caswell (1987) take up prioritizing as a step in applying a quality model. Nielsen (1993) notes in general the need of prioritizing, as do Dromey (1996) and architecture analysis represented by Svahnberg et al. (2003) and Svahnberg & Wohlin (2005).

Attribution

Each quality characteristic can be *attributed* to the information system as a whole or to some of its constituents. In the latter case it is thought that the quality in question is especially related to that particular constituent. Dromey's (1996) model includes attribution in the sense that each system component is related via quality carrying properties to quality attributes. Otherwise background theories do not clearly note attribution.

Definition

The actual *definition* of an attribute consists of the propositions relevant to understanding its meaning. It is a development of the initial general definition that is used in attribute set. All pieces of discourse relevant to the understanding of a particular quality attribute form the basis for its definition. Significant sentences are filtered out from the discourse, and thereafter propositions and predicates from sentences. A *proposition* is a sentence that is true or false, and *predicate* is a verb phrase template that describes a property of object or a relationship among objects.

Most of the traditional quality models attach definitions to their quality attribute lists. Nielsen (1993), for example, gives to usability and its components general definitions. He discusses the components and their measurement extensively and thus provides what can be called "discourse". McCall et al. (1977) and Boehm et al. (1978) and ISO 9126 (2001) give each characteristics a general definition as well. Grady and Caswell (1987, 157-159) only state that each term can be defined specifically for a project. The concepts 'proposition' and 'predicate' are not used in the background literature, but it would be easy to convert, for example, rows from ISO 9126 (2001) extensive metrics tables into propositions about the characteristic in question.

Use Case

Applying object-oriented terminology (Jacobson et al. 1999, 5, 41, 432), a *use case* is an interaction between the "owner" of the quality attribute, be it the system as a whole or some of its constituents, and the user. This interaction includes a specified sequence of actions, that the owner performs, and it yields an observable result to the user. Use cases are central to dealing with quality attributes. With respect to quality model use cases must be relevant for identifying and defining attributes and measuring their values. Traditional quality models did not develop any precise concept of 'use case'. Nielsen's (1993) 'tasks' can be compared to use cases. According to him usability is measured relative to users and tasks. Similarly ISO 9126 (2001) names 'task' as one of the factors that determine context of use.

Related to

The definition of a quality attribute *relates the owner of the attribute to things in its environment*. For example, 'usefulness' relates the application A to user N, task T, etc. These things can be as well human as non-human. Often these relationships manifest themselves via use cases and can be of varying complexity.

Models like McCall et al. (1977), Boehm et al. (1978), Grady and Caswell (1987), Nielsen (1993) mainly list, group and measure quality attributes, but don't discuss the nature of the attribute construct. Dromey (1996) like ISO 9126 (2001) in a way philosophize about the internal and external quality, but even these writings do not take up clearly and analyse the system-environment relationship. Jacobson et al. (1999), in turn, underline the importance of system context in general. The system-environment relationship is however implicit in most of the quality attribute definitions and measurement procedures and therefore needs to be explicit part of the quality meta-model.

Connected to

Quality attributes have interrelationships: they are *connected* or related *to each other* in different ways. The simplest relationship exists between the whole and its parts, or between group and its members. One attribute can be composed of other attributes or attributes can belong to the same category. Nielsen, for example, uses the terms "category" and "component" (Nielsen 1993, 25-26). Utility and usability belong to a category called "usefulness". Usability, in turn, has many components like learnability, efficiency, memorability etc. Accordingly, usability is a composite attribute, and learnability an elementary attribute or component. This means further that the learnability of the application, for example, affects the usability of the application. In addition to the category-component and a kind of influence relationships, Nielsen notes the trade-off relationship between attributes (Nielsen 1993, 25, 41). McCall et al. (1977) notice relationships between quality attributes as well as Boehm et al. (1978). Architecture analysis, for its part, discusses how attributes interact, support or compete with each other (Svahnberg et al. 2003, Svahnberg & Wohlin 2005).

Measurement, Value and Relativity

Measurement refers to the method of measuring the individual qualities or overall quality of an information system, i.e. finding out the current values for different quality attributes that have been attributed to the system in question. In the strictest sense *quality metrics* is rather a *part of information system development process model* than a part of quality model. A quality model, as it is understood in this study, specifies what to measure, not how to measure. This complies in a way with what already Grady and Caswell (1987, 2) state:

"software metrics program is part of the process of managing software development". Similarly Hevner et al. (2004, 78) differentiate, in their article about design science, evaluation process and method artifact from the product that is evaluated. However, because of the close relationship between metrics and quality model, metrics is to certain extent taken up in the study. Metrics has also commonly been an important part of quality models (McCall et al. 1977, Boehm et al. 1978, Grady and Caswell 1987, Nielsen 1993, ISO 9126 2001).

In connection with quality measurements it is important to differentiate between *actual* and *desired* values of an attribute. Nielsen (1993, 80) names desired values as "goals". In addition he separates current value from the *planned* value, and *minimum* value from best *possible* value. ISO 9126 (2001) states that it is not necessary to achieve perfect quality, but the necessary and *sufficient* one. Further, one and the same attribute can have both *absolute* and *relative* values. Execution speed of a function, for example, can be 'n seconds' or 'fast/slow'. The latter value has to be determined in relation to other processes in the context. In addition, attribute values can be *relative* in another sense too, namely relative to informants (users, etc.) or tasks (Nielsen 1993, 23, 27, 43). They get different values when measured by different actors or in connection with different tasks. In other words, quality measurements take place in a context which must be taken into account.

In the initial quality model *relative to* -element does not only refer to the relativity of measured attribute values. It applies to certain other aspects of quality model as well, like definitions and priorities. These were discussed above in section 4.1 in connection with attribute meta-model and called relativity points. ISO 9126 (2001) also notices different reasons for relativity: 1) user is not always aware of his real needs, 2) needs may change, 3) users have different operating environments, and 4) it may be impossible to consult all possible types of user. According to Nielsen (1993) usability is measured relative to users and tasks. Architecture analysis, in turn, points to the different views of stakeholders.

4.3 A short comparison to theoretical background

TABLE 8 summarizes how the constructs of initial meta-model are related to theoretical background of the study. Filled rectangles indicate that a model element can be clearly identified in and abstracted from the background theory in question. Rectangles with no filling and with dashed lines indicate that the idea is more or less implicit and between lines. The terminology in background writings might be different, but the concepts are exactly or almost the same as the ones used in the meta-models of this study. McCall et al. (1977), for example, use the terms "quality factor" and "criterion" for quality attributes, and "orientation" for what are called domain in the meta-model. Attributes are in some writings also called "properties", "characteristics" or just "requirements". In general different attribute groupings or categories are interpreted as wide or specific areas of concern, i.e. domains. As another example, what is meant by information system in the meta-model is sometimes called "software", "product" etc. What can be "identified" in the background theories, is sometimes difficult to judge in an absolute sense. Some aspect may have been referred to in the writings with one or two statements, but that has not been taken as an indication that the model in question really includes the aspect in question. The aspect must have been discussed more widely or be in some other way obvious.

TABLE 8 Origins of initial meta-model in theoretical background

	McCall et al. (1977)	Boehm et al. (1978)	Crady and Caswell (1987)	Nielsen (1993)	Dromey (1996)	Jacobson et.al (1999)	ISO 9126 (2001)	Svahnberg et al. (2003)
ACTOR				■		■	⬚	■
INF SYSTEM	□	□	□	□	□	■	□	■
ENVIRONMENT						■	■	□
QUALITY ATTR	■	■	■	■	■	□	■	■
QUALITY ATTR SET	■	■	■	■	■		■	■
DOMAIN	■	■	■	■	■		■	■
PRIORITY			■	■	□			■
ATTRIBUTION					■			
DEFINITION	■	■	□	■			■	
USE CASE				□		■	⬚	
RELATED TO					⬚	⬚	⬚	
METRICS	■	■	■	■			■	
CONNECTED TO	■	□		■			□	■
RELATIVITY							■	■

The meta-model has been created in order to overcome problems associated with the understanding of quality and corresponding models in theoretical background. One of the main points presented in section 3.1 above was that none of the models brings in attention all important aspects of quality and quality modelling, i.e. lack of comprehensiveness. By looking at TABLE 8, one can see that most authors and models recognize the information system and its quality characteristics, and present them as a categorized list (quality attribute set). Quality models - like McCall et al. (1977), Boehm et al. (1978) etc. – include also extensive description about how to measure the quality attributes. In addition, attributes are usually given a definition and their prioritization is recommended. What is, in turn, rarely set into focus are 1) explicitly attributing quality characteristics to specific system components instead of the system as a whole, 2) environment and system's relationships ("related to" in the table) to it, 3) use cases, 4) relativity inherent in quality models and 5) the variety of actors that can be involved in modeling quality. Consequently none of the background theories is alone comprehensive enough to cover all essential aspects of quality modeling. The initial meta-model clearly addresses this defect. An additional indication of meta-model's comprehensiveness is that it gives guidelines both for modeling overall quality and individual quality attributes.

Another point was that attribute lists and categories in traditional quality models are fixed, incomplete, ambiguous or even show a wrong hierarchy of characteristics. The initial meta-model solves this issue by not prescribing any set of attributes or domains, only the structure of general quality design. The development team can decide over relevant attributes, categorization and prioritization, and find out the attribute connections. Further, it was noted that there has been a tendency to equal quality to internal characteristics of information system. To address this problem the meta-model contains element "related to" and maintains that definitions of quality attributes actually relate the owners of the attributes to things in the environment. And by instantiating various elements of attribute sub-model it is possible to handle even fuzzy and complex system-environment relationships.

The next point in section 3.1 was that developer's view is easily dominant in quality models. The meta-model, in turn, underlines engaging different types of actors into quality modelling. Regarding suitable and agile enough metrics the meta-model again does not give any prescriptions. It only stakes out important differentiations, like between actual and desired value, minimum and maximum value, etc. Finally, the critics of theoretical background pointed out that the meta-level or ontology behind models is seldom discussed. As was noted in previous chapters, compared to the quality meta-model, traditional quality models (McCall et al. (1977), Boehm et al. (1978), Nielsen (1993), etc.) are mostly general level quality attribute collections. They name and categorize quality characteristics that information systems in general should possess, give general definitions to attributes and present techniques for measuring their values. In addition traditional quality models can take up and name some concerns, with respect to system life-cycle, or other aspects. All this happens, however, usually on a general or lower level. A typical example of general level concepts, in turn, is 'user'. A higher or meta-level construct would be 'actor' that covers both users and human actors in other roles that are relevant in quality modelling. Only a few concepts found in these models, like 'priority', can be positioned on meta-level.

The proposed meta-model goes one step higher in the abstraction to the so called meta-level by defining what elements or aspects any comprehensive system specific, system type specific or general information system quality model should contain and specify without prescribing the exact contents of these specifications. The meta-model tells, for example, that there must be a prioritized set of quality attributes grouped according to relevant concerns, but it does not prescribe and name what are the actual attributes, domains and priorities. In other words, the meta-model only addresses the ontology of quality models. It works on the level of information system product quality models in general.

5 APPLYING THE META-MODEL IN EAST-AFRICA

The previous chapter presented the first version of the quality meta-model and compared it to traditional models. This chapter relates how the meta-model was applied in three different system development projects. An overall description of African countries as information societies is given as background. The account of the research activity is divided into two parts. It starts with a description of the two first case studies that resulted in an intermediate meta-model. The second part relates the third case study. The descriptions are organized according to meta-model elements.

As was stated previously in the chapter 3 about research questions and design, the study is to a large extent inductive and theory creating. The traditional quality models can be treated as a kind of data representing general level quality modelling. The actual and fresh empirical data about quality modelling comes from the three case studies. This data is used to evaluate the usefulness of the modelling process and system specific models that are created on basis of the meta-model. The case studies help also to find out how comprehensible the models are for participants and how well they represent the real world. Further the comprehensiveness, generality and flexibility of the meta-model can be assessed on basis of the case studies. In addition, the definition of information system quality develops gradually out of the findings. Finally the case studies provide some clues about dividing the quality modelling process into steps and merging them into overall system development.

It is important at this point to remind the reader again about the fact that the system specific quality models created in the course of case studies must be characterized as prototypes. The researcher had to accommodate the quality modelling activity to the given conditions of the case study. In the first case one of the limiting factors was a very slow pace of system development process. In the two other cases the researcher had to share his time with quality modelling and other development activities within the limited timeframe of his assignement.

5.1 African countries as environment for information systems

According to Steven Alter's (1999) general theory of information systems the system environment includes the external infrastructure that an information system relies upon to operate and the managerial, organizational, regulatory, and competitive context that affects its operation. Infrastructure is further defined as shared human and technical resources, even though these resources exist and are managed externally. Infrastructure typically includes *human infrastructure*, such as support and training stuff, information infrastructure such as shared databases, and *technical infrastructure* such as telecommunications networks and programming technology. *Context*, in turn, includes other environmental factors that affect the system's performance. Intangible aspects of the context include the organization's culture, the current organizational and competitive climate, and the goals and opinions of various stakeholders. The relevant context may also include explicit rules and requirements such as government regulations, industry standards, and organizational policies. (Alter 1999)

The following description of Africa, and thereby Tanzania and Mozambique, as an environment for information systems follows Alter's differentiation between technical and human infrastructure. Context is discussed under other contextual aspects in section 5.1.3. The text is further organized along the World Summit on the Information Society (WSIS) targets (WTDR 2010). WSIS was held in Geneva (2003) and in Tunis (2005). In these meetings representatives of governments, civil society and business sector discussed subjects related to ICT. As a result ten targets to be achieved by 2015 were identified in Geneva. Most of the goals relate to technical infrastructure. A list of the WSIS targets (WTDR 2010, xxiii) grouped according to Alter's differentiation between technical and human resources is given below. (From technical infrastructure list the goals 7-9 related to television and radio services and to the use of world languages are omitted.)

Technical infrastructure:

1. To connect villages with ICTs and establish community access points.
2. To connect universities, colleges, secondary schools and primary schools with ICTs.
3. To connect scientific research centres with ICTs.
4. To connect public libraries, cultural centres, museums, post offices and archives with ICTs.
5. To connect health centres and hospitals with ICTs.
6. To connect all local and central government departments and establish websites and e-mail addresses.
10. To ensure that more than half the world's inhabitants have access to ICTs within their reach.

Human infrastructure:

7. To adapt all primary and secondary school curricula to meet the challenges of the information society, taking into account national circumstances.

Description of environment relates to general quality factors like security, availability, sustainability, affordability and maintainability, that are essential quality requirements for information systems targeted in the case studies. Environmental factors, in turn, act as barriers or bridges with respect to implementing ICT. Unless otherwise indicated the figures are taken from WTDR (2010). This ITU (International Telecommunication Union) report does not contain complete figures describing Africa and African countries as information societies, but general conditions characteristic to developing countries are applicable to Tanzania and Mozambique as well. Global and African trends and figures given in sections below are used as points of reference to make country figures understandable.

5.1.1 Technical infrastructure

According to WTDR (2010) among the most important worldwide developments since WSIS 2003 and 2005 has been the rise of mobile telephony and related applications. Secondly, Internet is considered as a general-purpose technology and access to broadband as a basic infrastructure like electricity or roads.

Mobile phones and Internet (target 10)

Mobile cellular network coverage stands already at 86 per cent of world's population and it is expected to reach close to 100 per cent by 2015. By the end of 2007 worldwide mobile cellular penetration stood at 67 per cent. Developing countries surpassed the 50 per cent penetration in 2008. In Africa the penetration rate was in 2009 32 per cent and it is predicted to grow up to 80 per cent by 2015. Subscription data obviously overstate the actual number of people having and regularly using a mobile phone for example because of duplicate, inactive or machine subscriptions. The majority of subscriptions in developing countries are prepaid and in Africa almost all are.

By the end of 2009 26 per cent of the world population were using the Internet. While in developed countries the penetration was 64 per cent, less than 20 per cent of people in the developing world were using the Internet, in Africa only 7.5 per cent. And only about 2.5 per cent of African households were connected to Internet. At the end of 2008 25 per cent of households globally had Internet access. In the developed world the rate was 60 per cent compared to 12 per cent in the developing world. In addition, the corresponding broadband penetration in developing countries was only 3.5 per cent. However, recent developments in the mobile sector are expected to have a major impact on wireless broadband access.

Internet connectivity is of great importance to the accessibility and maintainability of information systems. And in the era of cloud computing even possibilities of system integration are to some degree dependent on the Internet. In Africa the scarcity of broadband connections and surprisingly high subscription fees (and charged usually per amount of data transferred) have a hampering effect on system projects. In Zanzibar, for example, it was some times impossible to download applications and updates with a size of tens or hundreds of megabytes (today a normal size).

Rural areas (target 1)

The cost of installing wireless systems in rural areas is far less than the fixed lines. By the end of 2008 almost 75% of the world's rural inhabitants were covered by a mobile cellular signal. The highest rural coverage, 99 per cent, was in Europe and the lowest in Africa, just over 50 per cent (overall coverage in Africa is 69 per cent). The proportion of rural households with a mobile telephone has reached 50 percent in many developing countries. Figures for Mozambique and Tanzania are not available. Those African countries that are listed in WTDR (2010), however, show a penetration rate of only 40 per cent or under.

In contrast to mobile technologies, many rural households are deprived of access to Internet due to lack of electricity and high price of computers and connections. Especially broadband access is very rare. The proportion of rural households with Internet access at home exceeds 50 per cent only developed high-income economies like New Zealand, Israel and Japan. A growing number of developing countries are installing public Internet facilities in rural areas. Statistics about Tanzania and Mozambique are, however, not available. In general Africa has a very low level of rural Internet access. The data is, however, from between 2000 and 2006 and it is not possible to see the more recent changes. Another form of Internet access in both rural and urban areas is commercial facilities, in Africa so called cybercafés or Internet cafés.

A significant recent development in East-Africa has been the launch of SEACOM undersea fibre-optic cable in July 2009. Countries can link their fibre-optic backbones to the sea cable and extend high speed network (backhaul transmission networks) access to rural areas. In Zanzibar the link was realized in 2010. A parallel development is the spread of wireless broadband (3G standards) with several Mbit/s access.

Two of the case studies, website and educational statistics system, were carried out in urban areas, Maputo (Mozambique) and Dar es Salaam (Tanzania) respectively. The environment of land registration system, Zanzibar (Tanzania), can be regarded as a rural area. Users of all three systems are, however, spread all over the countries in question. Therefore, performance and the public availability of the services provided by applications are obviously affected by the general conditions of Internet access. Especially access from rural areas can be expected to be very limited. Commercial Internet cafés are common in Mozambique and Tanzania and the fees per hour reasonable for many citizens.

Schools (target 2)

Overall in developing countries Internet penetration is low and many schools are deprived of any form of Internet access. And even worse, many students do not have access to a computer at all. The New partnership for Africa's Development (NEPAD) e-School initiative was announced in 2003 for implementation until 2013 (WTDR 2010, 32). One of its goals is to equip all primary and secondary schools with ICT apparatus. Countries like Finland and Sweden have a computer for every 3-5 learners in public schools. There are no figures available for Mozambique and Tanzania, but generally developing countries have ratios with one computer for more than hundred learners. It is important to note that the above figures include computers that in practice are set strictly for administrative use.

Statistics suggest that by 2010 practically all schools in developed countries are connected to Internet. Data on ICTs in education is scarce in Africa. The few available figures suggest that the situation is not good at all. For example in Ethiopia and Senegal less that 10 per cent of schools are connected to Internet. Only in Northern Africa in countries like Egypt and Tunisia connectivity raises over 50 percent. The broadband access rate, however, is much lower. Broadband Internet access is essential to making use of Internet's full potential in education. Further, the use of ICTs is connected to the availability of electric power. Many schools in developing countries still don't have access to electricity. Again no figures are available for Mozambique and Tanzania, but for example in countries like Ghana, Senegal and Ethiopia only a minority of primary and secondary schools have electricity.

With respect to the case studies the status of ICTs in schools affects mostly the Tanzanian system for nationwide educational statistics. Schools are proposed to enter their statistics into the central national system and later download data for different purposes. But the aid organization's website could also be useful in schools. To help the situation the Tanzanian ministry of education equipped at the end of 2005 all regional and district offices with one computer and printer.

Scientific and research centres (target 3)

Most research centres and universities are connected to the Internet, often with a broadband connection. In some developing countries the major universities have acted as the first Internet service providers (ISP). By early 2010 around 62 per cent of countries had a national research and educational network (NREN), ranging from 100 per cent in Commonwealth of Independent States (CIS) to 33 per cent in Africa. NRENs are specialized ISPs with high speed backbone network dedicated to support needs of the research and education community. No detailed information is available about the situation in Mozambique and

Tanzania, only the names of NRENs, MoRENet (Mozambique) and TERNET (Tanzania). Land information or educational statistics are of interest to researchers, but from the viewpoint of the case studies, however, the status of NRENs in research centres has no significant effect.

Libraries and cultural centres (target 4)

Libraries, post offices and cultural centres are ideal locations for providing public Internet access to the community. Most developing countries lack the resources for providing ICTs for cultural centres. Bandwidth is also a constraint together with connection charges. The lowest figures come from Africa from 2009, where 71 per cent of countries reported that less that 20 percent of public libraries offered Internet access. Post offices are discussed together with cultural centres in WTDR (2010). Statistics show that in Tanzania less that 10 per cent of post offices provide public Internet access (PIA). In Mozambique the figure is even smaller, just a few per cents. The three case studies do not touch this sector of information society, nor the health sector discussed shortly below.

Health institutions (target 5)

By the end of 2009, some progress had been made in establishing basic Internet access in health institutions. These institutions are also increasingly using ICTs for their own ends. Most countries have some form of electronic patient records, but in low-income countries most records are still kept in paper format. About 75 per cent of low-income countries report at least one m-health (health practice supported by mobile communication devices) project. In South Africa, for example, mobile technology is used to provide home-based care to HIV/AIDS patients. Another project of importance to Africa is HINARI (Health InterNetwork Access to Research Initiative) with a mission to provide online access to leading biomedical journals for non-profit health related institutions in developing countries. Percentage of total institutions connected to the initiative in Africa is 35, the largest in world.

No statistic is available about Mozambique and Tanzania, but in general in African developing countries the percentage of health institutions with access to internet is very low. Similarly the patient information is primarily kept in paper format. However, even high-income countries experience difficulties in transition to electronic records and electronic transmission.

Public sector (target 6)

Many countries have been reforming and modernizing their public-sector systems. This includes putting in place ICT infrastructure and promoting the use of ICTs. WTDR (2010, 120) report formulates the benefit of e-government as follows:

> "The 24/7 availability of ICT tools enables faster and more efficient communication not only between the public sector and citizens, but also between government agencies, processes and systems. Thus, ICTs help governments operate more efficiently, improve the delivery of public services and enable more widespread information dissemination to citizens."

In general the percentage of government entities with internet access is comparatively high. At the end of 2009 189 countries had a central government website. Also many government ministries and departments had a web presence in the form of a sectoral website. National home pages have grown to integrated portals. On the local level development has been slower. By 2008, however, more than half of all countries had one or more local government sites. According to one study (Holzer & Kim (2007)) 50 per cent of the cities selected in Africa had an official website in 2007.

It is important to note that just setting up a website is not a sufficient condition for effective e-government. The site must contain useful information and offer online services. Especially developed countries are providing sophisticated interactive services. According to 2008 statistics online form submission is the most common functionality. It was reported by 39 countries worldwide. Online payments were possible in 31 countries. By 2009 21 countries out of 192 worldwide offered tracking of permits as an online service to their citizens. In developing countries lack of resources – financial, human and infrastructure – is a constraint for increasing access to the Internet.

An important aspect of e-government is citizen's participation in public policy-making. Implementations in this area made for the most part in developed countries. 28 per cent of surveyed countries have some kind of statement encouraging citizen participation, 11 per cent provide a mechanism for consultation and only 9 percent give some feedback to the citizen.

The status of e-government in Mozambique and Tanzania cannot be seen from the available WTDR (2010) statistics. National websites exist, but the more advanced e-government features are probably missing or in very initial state. In addition many sites are often down, out-of-date or poorly functioning. The public sector ICTs constitute, however, part of the environment for two of the case studies carried out in Tanzania, the first dealing with education statistics and the other one with land registration. Both systems are hosted by respective ministries and their departments, and contribute to the e-government in the country. Starting 2011 Zanzibar is building a fire-optic backbone for government authorities. This, as well as exploitation of 3G mobile technology for entering field data, will boost the use of the land registry system. Government websites are excellent means to make publicly known and available both educational statistics and land information. More advanced services, at least form submitting, are needed in these two cases.

Availability of hardware and software

There is no locally manufactured ICT equipment in Tanzania. Practically everything is imported through private companies and international manufacturers. A limited number of local software and website development companies however exist. An interesting historical fact is that in early 1970's Tanzania banned importation of information technology. Later in 2000 government removed the value added tax on computers and peripherals. As a result currently these technologies are widely available at least in big cities. Second hand computers can be bought for about 150 USD. This is, however, still too expensive for poor people in rural areas. (Yonazi 2009) Despite of fairly good availability of domestic hardware and office software, delivery and service times can still be long when it comes to more advanced technology like database management systems and server machines. This was experienced in connection with the both Mozambique and Zanzibar cases. It inevitably affects the maintainability of information systems.

Electrical grid

Information technology is dependent of steady supply of electric power which is not self-evident in East-Africa. In Tanzania, for example, power rationing occurs on regular basis. In February 2011 Thomson Reuters news published an article according to which Tanzania Electric Supply Company (TANESCO) could generate only two thirds of demand due to prolonged drought. Most of Tanzania's electricity is hydropower generated. More than fifth of Tanzania's generated electricity is lost during transmission and distribution. While only 14 percent of country's 40 million people are connected to the grid, the demand is growing by 10 to 15 percent annually. (Anon 2011) Rationing means sometimes daily power cuts of several hours.

In addition to scarce supply, power cuts can occur for other reasons too. During researcher's stay in Zanzibar the sea cable bringing electricity from mainland was damaged two times. First occasion in 2008 caused a one month long power cut and the second one in 2009-20010 a three months brake in power supply. Sometimes power lines are damaged by thieves who are after copper used in wires. During these times people must resort to their own petrol or diesel generators. Ordinary citizens, however, cannot afford that kind of equipment. In addition to total power cuts, disturbances like voltage spikes are common. After the three months cut in Zanzibar when power came back to the grid, researcher measured 400V from the sockets at home! All in all, the uncertainty in power supply is a big threat to the safety and availability of information systems. Hardly anything can be set up without UPS apparatus, surge protectors, etc.

5.1.2 Human infrastructure

ICT education (target 7)

Guaranteeing an adequate supply of trained teachers remains a major challenge in both the developed and developing countries. The problem with WTDR (2010) statistics is that the indicators are limited to about 20-25 economies.

Percentage of teachers who have ICT qualifications in primary and secondary schools varies from 0 to 10 per cent in countries with available data. This does not, however, necessarily mean that these teachers actually teach ICT skills.

Regarding ICT-assisted instruction only a small proportion of schools in developing countries have effectively integrated ICTs as part of the curriculum. Proportion of primary and secondary school teachers trained to teach their subject using ICT ranges from 0 to 100 (WTDR 2010, 145). In general the level of computer-assisted instruction is higher than the level of Internet-assisted instruction. In some developing countries, however, a large proportion of schools practise Internet-assisted instruction (WTDR 2010, 145). Again no statistics is available in WTDR (2010) about Mozambique and Tanzania.

The status of ICT education itself in Tanzania at the time of the first case studies is described by Vesisenaho (2007, 41-42). According to him by 2004 four institutions were offering computer science or information technology education. The largest of these institutions is University of Dar es Salaam which started to offer a masters level computer science degree already in 1974. Some ICT training components have also been added in teacher education. The total enrolment of ICT students in all the four institutions was under 900 in 2005.

How ICT skills are taught in primary and secondary schools and on higher levels together with the number of ICT professionals coming out of educational institutions affects the sustainability, maintainability and use of information systems in public and private sector. The situation is fairly good in big cities like Maputo and Dar es Salaam, but in Zanzibar the lack of human resources in this respect is noticeable. Another factor that worsens the situation is low salaries for ICT professionals, especially in governmental organizations. A person hired will often soon leave organization because of better salaries offered elsewhere in the country or outside the country.

5.1.3 Other contextual aspects

National strategies

According to WHO Global Observatory for eHealth 2009 survey 72 per cent of low-income countries have a national e-government policy, 38 per cent of them an e-health policy and around 20 percent telemedicine policy (WTDR 2010, 110). Tanzania instituted its national ICT policy in 2003. It has 10 focus areas, among them infrastructure, industry, legal and regulatory framework, public service, local content and universal access. The implementation has, however, met problems. There has been no central coordination and no agreed implementation strategy. In general the infrastructure layer has been promoted at the neglect of other layers. Telecommunication market became fully liberalized by 2005. A new ministry of Communications Science and Technology was established 2008. It has the responsibility of developing policies particularly in the area of ICT. (Anon 2010, Yonazi 2009)

Finance and economy

Tanzania is one of the poorest countries in Africa in terms of GDP (Gross Domestic Product), number 158 of 180 countries in IMF ranking 2008. Around half of the population lives below the absolute poverty line. (Anon 2010) Against this background it is easy to understand that governmental and other non-commercial organizations lack financial resources to sustain use of information technology. They are heavily dependent on foreign aid. Accordingly sustainability is too often at stake when development co-operation projects end.

Organizational and cultural aspects

Krishna & Walsham (2005, 134-135) take up five common features of developing country context that bear on system development. Four of them pertain to organizations and culture.

- The extent of reorganization needed in organizational systems and processes for information system objectives to be met is much higher than in developed countries.
- Staff, especially in governmental organizations, cannot be hired or fired easily because jobs are scarce and social security is non-existent.
- Processes cannot be changed easily because of vested interest. High political involvement and commitment is needed.
- Due to very hierarchical social systems sometimes even the top leaders of a country need to be personally involved in information systems issues.

The above features were visible in the context of the third case study in Zanzibar where 95 per cent of the population are Muslims.

Climate

In an article about appropriate ICT for developing countries van Reijswoud (2009, 5) states that “a computer setup that is to operate in the African desert is not considered to be appropriate when it is not well protected against heat, sand and dust”. These kind of climatic conditions prevail not only in deserts but all over the tropic. To heat sand and dust one must add humidity. Electronic devices don’t necessarily stand a continuous high humidity. In Zanzibar, for example the relative humidity is in average around 80%. About the same figures hold for Dar es Salaam. Condensation and rust easily follow with humidity. High levels of dust particles in the air, in turn intrude the electronic equipment. When the researcher left his laptop after two years usage for service in Finland, the technicians told they never had see so fine-grained dust inside a computer. Finally, with the climate come different animals. Small ants are found everywhere, inside and outside houses. They intrude computers like the dust. Rats are another harmful species. Employees of the Ministry of Education in Dar es Salaam reported cases where rats had bitten cables and paralyzed the local area network. All this is a substantial threat to the safety, maintainability and availability of information systems.

5.2 The first case studies: EMIS and website

5.2.1 The Cases

The two first quality modelling case studies are discussed together because they were carried out parallel and influenced each other.

EMIS

The context of the first quality modelling case study was the development of a new system for managing educational statistics called Education Management Information System (EMIS) at the Ministry of Education and Culture (MoEC) (later Ministry of Education and Vocational Training (MoEVT)) in Tanzania. The goal was to replace an old system consisting of a COBOL database application and a set of Excel files. The system was used to produce the yearly Basic Education Statistics in Tanzania (BEST). The data was also disseminated to National Bureau of Statistics (NBS) and UNESCO. The project had a long history. Preparations for building the system were started already in 1997 together with foreign consultants. This first development cycle was ended by 2001 resulting in a small prototype that was never used in production. Thereafter a new round of planning began. Finally, in 2005 the project plan gained administrative approval and the actual development work started at the beginning of 2006.

A local software company was hired to design and program the system. A rare arrangement in African context as was noted above. The actual work was characterized by tight funding and unrealistic schedules. Nevertheless, a first system version was installed and put into operation during May-June 2006, after only three months work. At the beginning of 2007 the data model of statistical database was amended to accommodate some additional facts gathered with revised data collection forms. Later in the same year UNESCO took over the system development by importing its own solution.

Research and quality modelling activities were harmonized with the actual development activities of the EMIS. In general researcher's work consisted of being present at the ministry whenever something related to the development of EMIS happened and writing notes on observations. He also prepared extensive status reports, participated in training of end users and even helped in entering some of the raw data into the database. This can be characterized as participant observation. Parallel with all this the researcher applied the quality meta-model under study in order to create a quality design for the EMIS. All model elements and the composite quality model were considered together with selected informants using interviews or group discussions. The data produced and gathered is mainly qualitative. The only quantitative method used was AHP in connection with prioritizing quality attributes. The case was closed in June 2007. At that time the development cycle with the local software company ended and the researcher himself was fully occupied with another project in Zanzibar.

Website

The second parallel and assisting case study took place in Mozambique. It dealt with creating an initial version of a new local website for a global non-governmental organization, Lutheran World Federation (LWF), in Maputo. Besides this the researcher took care of a number of infrastructural issues at the office. The research component consisted of creating a general quality design for the website and modelling two top qualities in detail. As in EMIS case the research activities were harmonized with the other system development activities. Unlike the first case the researcher was at the same time in the role of developer and the communication problems experienced in connection with EMIS could be avoided. On the other hand the developer role consumed a lot of energy and contrary to expectations hampered the quality modelling effort.

The work started with an assessment visit to Maputo in October 2005. A report was written on basis of the visit about the current state of the information system at program office, problem areas, development needs and priorities. An action plan was embedded into the report indicating the next steps to be taken with respect to typical IS-project elements. Later on, at the end of November 2005, Maputo programme office categorized and prioritized development needs. Thereafter technical staff at Mondlane University was contacted in order to install a website prototype developed by the researcher. It proved, however, to be more difficult than expected. Most of the time the staff were not available, server platform renewal was going on, etc. Finally, at the beginning of May 2006, researcher decided to visit Maputo and push the work forward, and as a result the website prototype was installed and the development activities could really start. The work was carried out between May 2006 and June 2007. All preparatory work as well as most part of the programming work was done in Tanzania. All other activities were carried out in Maputo. The case was closed as researcher's work assignment ended and no other funding was available to continue the study.

5.2.2 Workflows and steps

The quality modelling process inside both case studies was carried out according to following general steps listed below (exceptions are indicated). The steps equal with meta-model elements. The work on steps 1 – 6 was more intensive than on the last three steps.

1. Identifying stakeholders and selecting informants who where used as source in subsequent steps.
2. Building overall understanding of system and its environment.
3. Creating an overall quality design for the system by selecting and prioritizing quality domains and attributes using a questionnaire, workshop and analytic hierarchy process (AHP), and finally attributing individual qualities to lower level system constituents if relevant. AHP was used only in EMIS case. In website case domains and attributers were prioritized separately.
4. Identifying use cases and scenarios relevant to selected quality attributes.
5. Eliciting discourse - using interview or group work - that contains explanations of what would it mean that the system, when it is ready and in operation, really has each of the selected qualities. Sifting out propositions from the discourse.
6. Identifying the relationships between the system and its environment that pertain to the selected qualities.
7. Identifying initially some system components or features that determine for their part if the system has or has not the selected qualities.
8. Identifying connections between attributes. Not modelled in website case.
9. Measuring the quality of target system during testing and operation. In website case metrics was omitted because the website was not yet published when researchers work ended.

FIGURE 15 depicts the quality modelling process (a process model) divided into main activities or workflows (rectangles) and shows their relationship to meta-model elements (numbered in diagram) and to the main workflows (see for example Jacobson et al. 1999, 11) of overall system development. The latter are named on the upper row in the diagram and separated by dotted vertical lines. Generally the process was incremental and iterative rather than of "waterfall" type and happened along with other system development processes. This is indicated in the figure with a narrow dashed line under the workflow rectangles and arrows joining the workflows. From any of the activities one could go back or forth to some of the other steps in order to refine other design elements. Steps 1 and 2 belong to an inception phase when actors for quality modelling are selected and understanding of system and its environment is built. Next two major quality modelling workflows are overall and detailed quality designs. The former deals with model elements 3a, 3b, 3c and 8, and the latter with elements 4 - 7. Contributors constitute a new meta-model element discovered during the study. All the above steps were carried parallel to general requirements gathering, analysis and system design workflows. Implementing quality design and measuring (step 9) the quality of resulting system were the weakest (symbolized by long dash dot line) links in the case studies. This was due to limited time frames and other resources, and to the view that metrics are not a genuine part of a quality model. Quality modelling could be carried out only during the design and implementation of first basic system components. None of the systems was fully complete at the time of closing the case. This complies, however, on the other hand with the research design where focus is set rather on modelling than implementing and measuring quality.

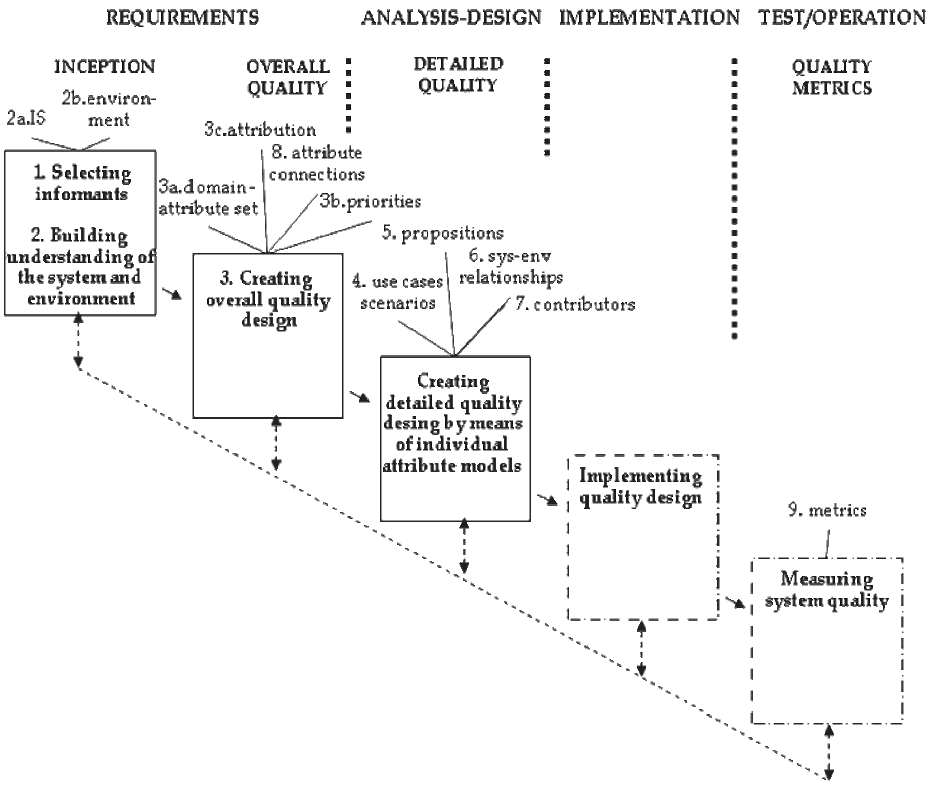


FIGURE 15 Quality modelling activities and system development workflows in two first case studies

5.2.3 Model elements

The detailed description of quality modelling during case studies and findings thereof is organized according to elements of initial information system quality meta-model. Main observations, findings and implications are highlighted with italics.

Actor

In the EMIS-case the number of potential stakeholder categories and consequently informants was very large in size and all of them could not be interviewed or taken into account otherwise. Therefore, a method known as "purposeful sampling" was used. It meant selecting eleven easily accessible informants with maximum variation on job and interest. The selection included persons representing ministry's IT and statistical staff, district level educational officers, related organizations (Agency for the Development of Educational Management (ADEM), Teachers College), national bureau of statistics (NBS), other ministries (Presidents Office – Public Service Management) and the international donor group. This informant group was used to deal with the quality attribute set, domains and priorities. After selection of the most important qualities the work was continued with the personnel of ministry's EMIS and Statistics Unit and a representative of the local software company that was selected to develop the initial system version. FIGURE 16 depicts the main central and local administrative bodies in Tanzania. In indicates how EMIS statistics is collected from district level departments to the central system. LGA stands for Local Government Authority, a common naming of district level administration. The organizational structure of Ministry of Education and Culture is depicted in APPENDIX 2.

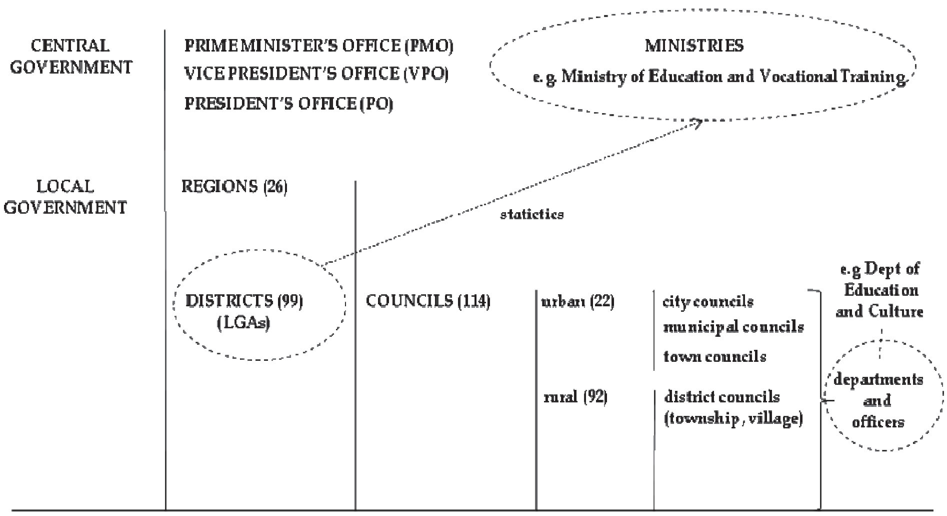


FIGURE 16 Main central and local administrative bodies in Tanzania

In website-case the informants were throughout the study members of a small development team chosen from employees of the local head office of the host organization (FIGURE 17). Among them were the country representative, administrator, assistant administrator, program coordinator, financial manager, logistics officer and two project officers. This selection can also be characterized as a purposeful sampling. In both cases the researcher tried to act as an editor of the quality model and not to influence the selection and definition of quality requirements. All the informants were affected by the system under development as current or future users and therefore were highly qualified for giving information related to system quality. They also represented the minority of population having access to computers and Internet and using these tools in the daily work.

FIGURE 17 Members of LWF website project team meeting in Maputo office

Both *case studies gave support to the importance of actor selection as a model element*. Because of the relative nature of quality (discussed below in section about measurement, value and relativity) stakeholder views have an impact on attribute sets and individual quality attribute definitions. It was seen in the big variance in individual domain and attribute prioritizations as well as various propositions describing one and same quality attribute. Consequently missing stakeholder views affect the defined and achieved overall quality. Involving different groups is especially important in the context of developing countries (Krishna & Walsham (2005, 137)). In the ideal group of informants all essential actor and stakeholder categories are represented. The studies demonstrated, however, how *difficult* it is *to contact and to engage a sufficient number of informants from all actor-stakeholder categories* into quality modelling. Therefore, instantiating a complete quality model in this respect is seldom possible in practice. In the website case a weakness was that ordinary users, who are not employees of the site owner, were missing from the group of informants. The case study quality models do, however, reflect a satisfactory number of different stakeholder views.

According to the initial meta-model actors and informants are defined as part of the attribute meta-model. This reflects the "attribute bias" in the first model version and the fact that the model was constructed solely on basis of initial literature review and the general requirements documentation of EMIS without a connection to actual quality modelling process and real actors. Very soon it was, however, realized that *human actors are* "ubiquitous" and *needed in every phase and step of quality modelling. In the visualization of meta-model actor-informants element must therefore be placed outside and on top of all other elements* to reflect this fact. And initial *group of actor-informants must be selected as the first step in quality modelling* and given the task of describing the system and its environment at the level appropriate for that stage of quality modelling.

Non-human actors were not discussed as part of actor element. Non-human actors cannot take part into the definition and measurement of qualities. *It was felt more natural to deal with them* in connection with use cases and external relationships (RELATED TO –element in the initial model) and *as a subset of entities in the environment*, namely related information systems. The initial definition of actor element was influenced by the notion of 'users' in unified software development process (Jacobson et al. 1999, 5).

Information System

Excluding user interfaces, the information systems in both case studies appeared more like black boxes or something behind a closed door (FIGURE 20) to most informants. Systems were not presented to them as a detailed and well defined set of constituents, their relationships and behaviour. This was due both to the lack of architectural descriptions and the disinterest of most actors in technical descriptions. A few diagrams were drawn, however, for technical reports. FIGURE 18 shows the overall system diagram of EMIS case. It shows on a very general level the main hardware and software elements of development platform. MoEVT stands for Ministry of Education and Vocational Training.

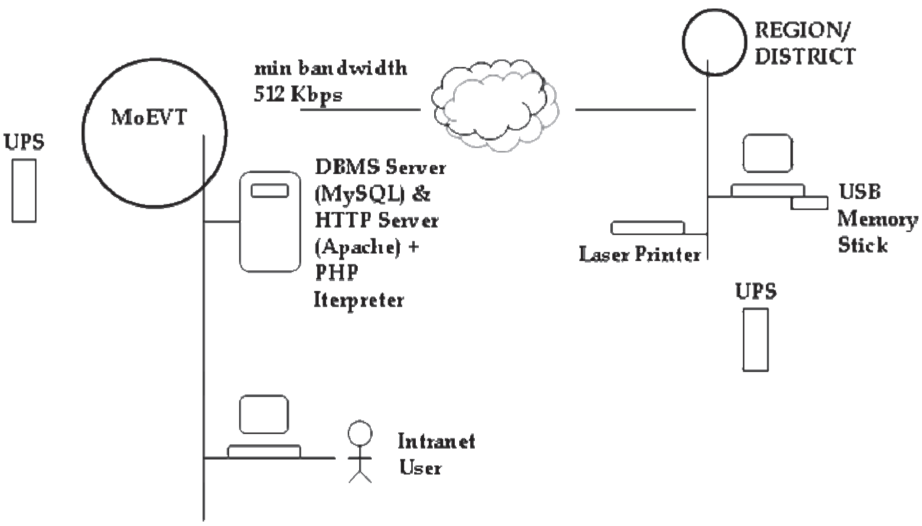


FIGURE 18 Overall EMIS system diagram

In the website case only a sitemap (FIGURE 19) was known to the informants in connection with interviews. They could, in addition, browse the test site that was running on the office server.

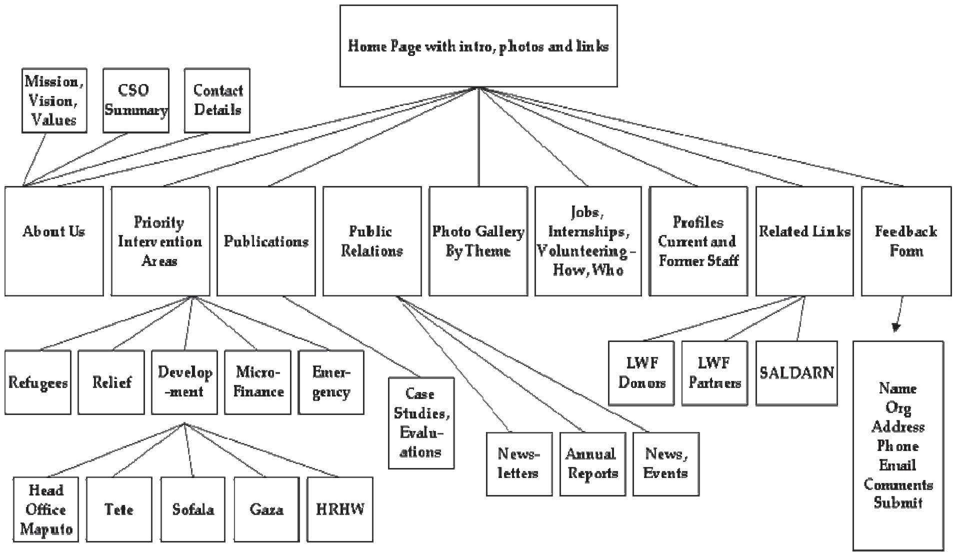


FIGURE 19 Site map of LWF Mozambique country office website

The *high level* almost *black-box view fitted*, in the end, *well to the emerging understanding of quality attributes as* rather *referring to relationships* (discussed below) *between information system and its context* than to inseparable features of the system itself. In a black-box or nearly black-box view there is little that distracts actors' attention too early from these important relationships. And even later, when it was time to go a bit more inside the system, the researcher tried, in order to clearly understand each quality requirement, to keep the *system internals* initially disconnected from environment and to look at them, just as plain constituents, structures, behaviour and pieces of contents, and *to give them in advance as little external justification as possible*. Seen in this way, an *information system is* in the course of quality *modelling described gradually from top down* starting from high-level architecture, then according to needs going down to design patterns, layers and packages, procedures and methods, records in the database, series of instructions, etc.

FIGURE 20 EMIS office and server room (behind the closed door on the right side)

Environment

The environment element is part of initial meta-model, but without any structured or detailed definition. The element was also present in case studies, for example in attribute collections as domains like “risk, harm to environment”, “change of environment” or “impact on environment”. Environment elements were also depicted in system diagrams. Environment was in the minds of actors broadly divided into technical and social environment. Further it was understood that a specific part of context like a user, user’s organization, business processes in organization or even the whole society can be put in focus at a time. Things like human and technical infrastructure, even climate, were felt especially important in Africa.

With regard to the importance of system-context relationships in quality modelling, the case study environment descriptions were, however, all too thin. And it was realized that the concept of environment covers actually so large number of different entities that some kind of *“environment meta-model” is needed to give structure and consistency to environment descriptions*. FIGURE 21 shows a simple diagram used in website case. It depicts the four main elements that interface with the website. Three of them are groups of human actors and one a group of information systems (other websites in this case). LWF stand for the name of host organization, the Lutheran World Federation. Another diagram was also drawn to show in more detail the different internal and external website user categories.

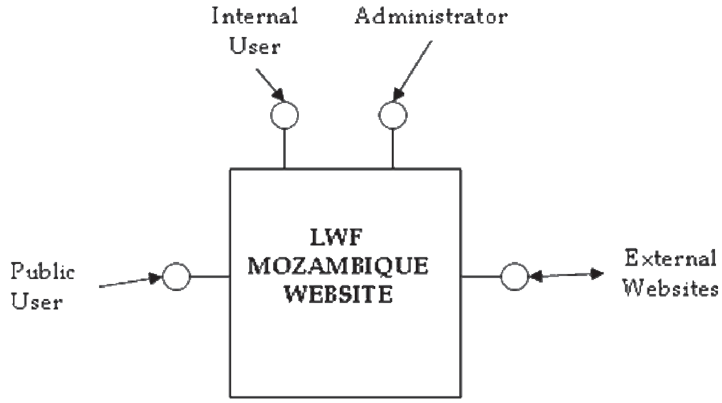


FIGURE 21 Website system environment

FIGURE 22, in turn, shows a diagram that was used in EMIS case to remind of important environment elements. MoF stands for Ministry of Finance, PO-P&P for President's Office Planning and Privatization, NBS for National Bureau of Statistics, TSED for Tanzania Socio-Economic Database, VETA for Vocational Education and Training Authority, NECTA for National Examinations Council of Tanzania. Both diagrams indicate how important external users, user organizations, connected systems, and input output data were taken into account.

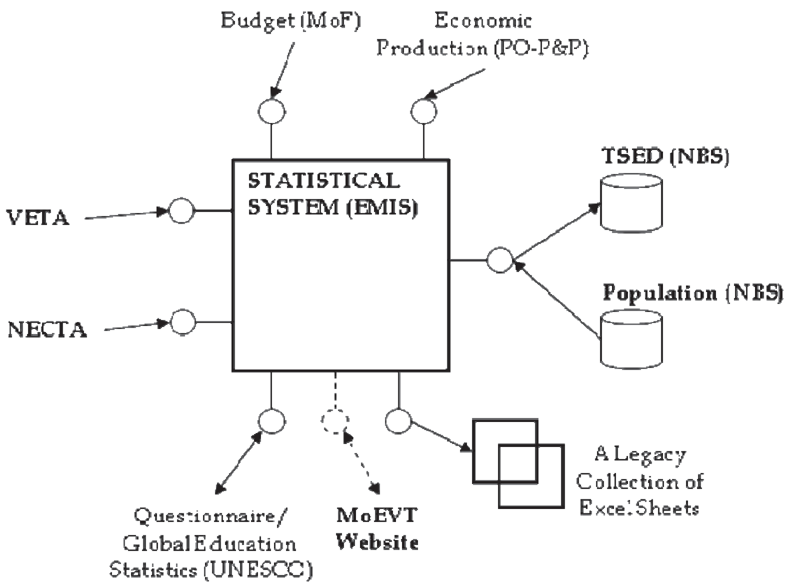


FIGURE 22 EMIS system environment

Outside the actual written or drawn environment descriptions the general African conditions were well known to all actors as well as their impact on security, safety, availability, sustainability, affordability and maintenance. Account of technical and human infrastructure and other factors was given above 5.1.

Domain, Quality Attribute and Attribute Set

If ordinary, non-ICT-professional informants are asked to list, for example, ten different qualities of information systems, a couple of first attributes may be easily found, but then difficulties tend to arise. This holds, of course, even more true in a context where the history of information technology is relatively short and only a smaller part of population have had access to it. Therefore in both case studies a predefined collection or set of domains and quality attributes was given to informants as a starting point.

In the EMIS case study the attributes were first emptied out of the domains of a general collection of quality attributes presented by Finne (2005) (see section 4.1 above). Then the domains were filled by the researcher with qualities mentioned in the EMIS development plan. The result thus constituted a system specific domain and quality attribute set that represented the “de facto” general quality design made earlier by the ministry. One new domain, “impact on environment”, was added to it as noted in Finne (2006) and the collection was then used in a questionnaire to assist selecting the most important domains and the most desired qualities. In both case studies general definitions of domains and attributes were created by the researcher and given to the informants as part of questionnaire. TABLE 9 contains examples of general definitions taken from the glossary used in the case study. The entire glossary can be found as APPENDIX 5.

TABLE 9 Examples of initial EMIS domain and attribute definitions

DOMAIN	ATTRIBUTE	DEFINITION
Data, information, contents		This domain embraces simply all kinds of contents (database records, text, pictures, video, documents, etc.).
	Reliability	The information in MIS is likely to be correct.
	Accuracy	The data in MIS is correct even in small details.
Risk, harm to IS		Different things can cause physical harm to the information system. They include human actions as well as other factors like electricity surges. The harm may appear in the form of loss of data, lost network connection etc.

FIGURE 23, in turn, depicts the whole domain-attribute set as it was presented to informants in EMIS case study. Attributes sustainability and suitability were positioned in the middle of the "palette" to underline their composite nature. This was later found as a mistake, because informants tended to neglect them. Attribute name "security" in the domain "harm to IS itself" was replaced with "safety" in the course of work.

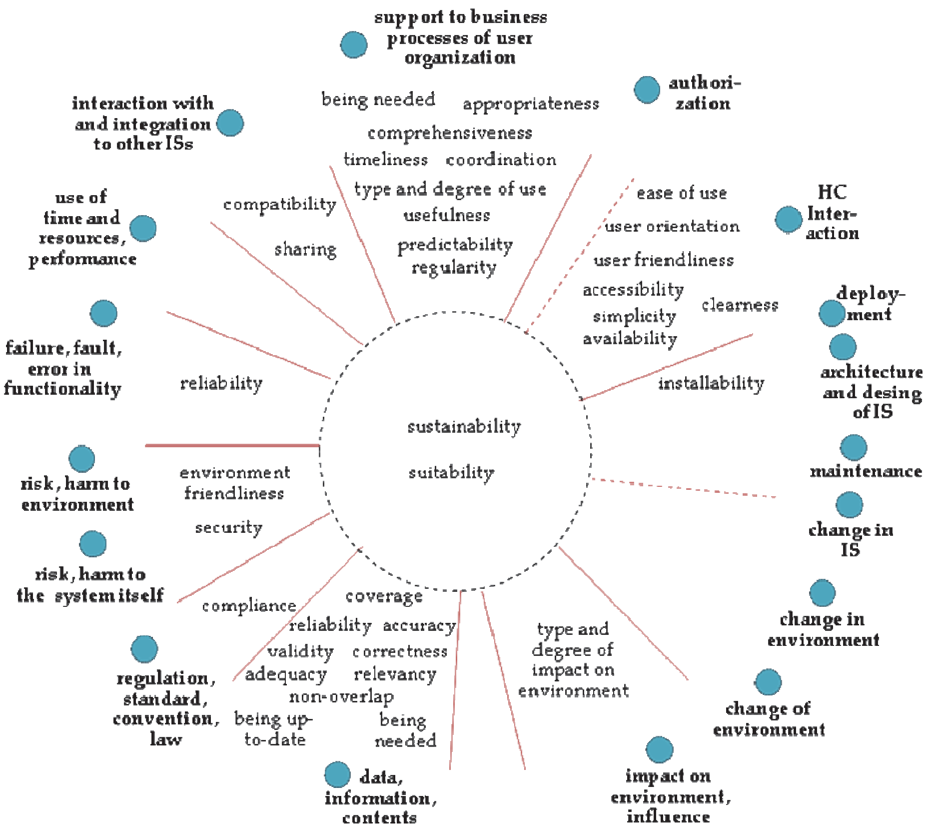


FIGURE 23 EMIS domain and attribute set

The quality attribute set in the website case was based on EMIS attribute set but it was modified together with the informants to meet some specific needs of a website. It included attributes like 'ease of finding information', 'download speed', 'ease of updating', 'browser compatibility' etc. Some new domains were also added. FIGURE 24 depicts the domain and attribute collection used in Mozambique. In this case only domain definitions (APPENDIX 6) were given to informants in the questionnaire.

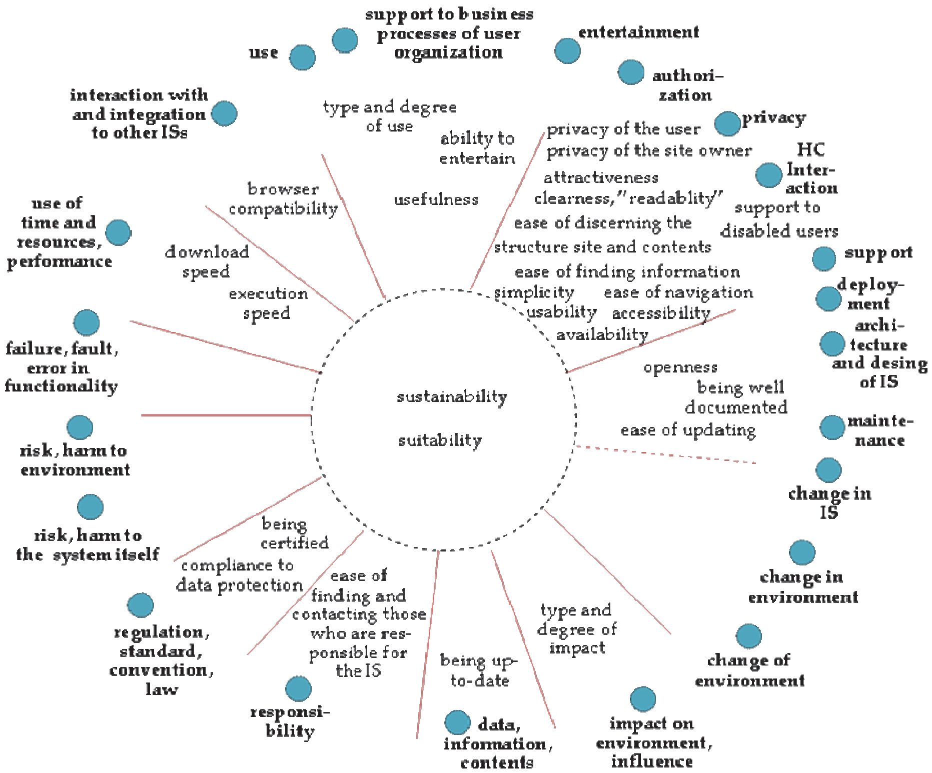


FIGURE 24 Website domain and attribute set

Compared to traditional models, which easily lead to fixed attribute sets and rigid categorization, the *collections in both case studies*, despite being predefined, were intentionally kept *open to new domains, new attributes and regrouping or even removing of attributes*. The principle of giving actual system development or quality modelling teams the final authority to define relevant domains for categorizing and selecting attributes gives flexibility to quality models and makes the meta-model fully applicable in different contexts. This approach is supported by articles published in recent years which take up new attributes that are not included into traditional quality models (e.g. Voas (2004), Voas & Agresti (2004)). Even new domains emerge. Change, for example, is noted as an important field of concern e.g. by Berki, Georgiadou & Holcombe (2004) and Bollinger, Voas & Boassson (2004). It relates to maintainability and modifiability. Main categories of change are: change in information system, change of information system, change in environment and change of environment. The latter means moving or installing information system into a new technical or social environment. Another two new domains are extensibility and security (Siakas & Georgiadou 2005). An additional good reference point is a model used by Scholl et al. (2011, 793) that lists values like individualization, aesthetics and stimulation. In general, however, a comprehensive and acknowledged theory of domains and attributes relevant to different information systems and information system development is still missing. Problem of categorizing, for example, has been noted by Svahnberg & Wohlin (2005). In addition to the emerging attributes and the set of requirements for a website is clearly different from the set of requirements for a statistical MIS. This was clearly reflected in the need to modify the domain-attribute set for the website case. The different importance of quality attributes to different software categories was noted already by McCall et al. (1977). Accordingly, both information system type specific and generalized attribute collections can be used to support early phases of quality modelling.

All in all, the *predefined domain and attribute collections* used in case studies *helped to identify, name and understand individual quality characteristics as well as disclosed biases and gaps in the existing (EMIS) quality design*. Totally neglected areas by the authors of EMIS development plan were 'authorization', 'architecture and design', 'maintenance', 'change' and 'performance'. Nothing was done, however, by the developers to fill them. The most populated domains in the same plan, in turn, were 'data, information, contents', 'HC interaction' and 'support to business processes'.

It was noted in both case studies that *actors need guidance in order to fully understand the terms used to name domains and attributes* in the predefined collection. Notions of attribute, domain and collection themselves were regarded to be clear and understandable. But in both case *studies the mere number of domains and attributes presented in collections was felt by some actors to be large to deal with*. Especially in connection with large information systems with many stakeholders, the number of quality attributes can grow very big. *This justifies splitting one general quality attribute set into a number of smaller more specific sets*. Several alternatives exist: differentiating between hardware and software qualities, between the central system and local systems, between separate subsystems or between the most populated domains (data, support to business processes and HC interaction). A synthesizing workshop or discussion can also correct possible misinterpretations, and give opportunity to finalize the general quality design.

Priority

In EMIS case the overall selection and prioritizing of the most important quality attributes was carried out by giving 11 informants a questionnaire that was collected and finalized together with the researcher in an interview. The questionnaire itself had two parts. The first one was a page for collecting basic information about the interviewee, and the second one a domain-attribute collection (FIGURE 23 above) for indicating opinion about the priority of domains and quality attributes. The informants were asked to select at least ten domains and attributes that the system under development should have and order them according to importance using numbers from 1 to 10 (or to the total number of selected attributes). Number 1 indicates the highest priority. Many of the informants in EMIS case did not orderly rank attributes because they thought that by prioritizing the domains they were also prioritizing the attributes on the same palette. Accordingly, *it was found easier for the informants to think first only about domains and thereafter deal with individual attributes*. It is also more logical first to think about areas of concern and then to address them with quality requirements. Based on this experience of the EMIS case, the palette was divided in the website case study into a separate domain collection and a combined attribute and domain collection (FIGURE 24 above) that were adapted for websites. In this case only 5 actors gave their opinion.

FIGURE 25 shows the five top priority domains of two first case studies. The limitation in number of top domains and attributes is based on cognitive and practical reasons (cf. Miyoshi & Azuma 1993, 426). The height of column corresponds to the inverse average ranking of domain in question (the higher the column the higher the priority). Because some informants were able to order even 15 domains the lowest priority value came to be 15. Therefore all domains not ranked at all in a questionnaire were given this lowest value 15. Inverse average in the resulting statistics was calculated by reducing the average value from 15. Columns with fill represent EMIS case and columns with no fill the website case. The entire results concerning domains are shown in APPENDIX 8 and 9 respectively. Only two of the top five domains are common to both cases: support to business processes and data-contents. This can partly be explained by the different type of the systems under design, in the first case a statistical MIS and in the second a website. FIGURE 26 shows a similar view of top five attributes. The entire results concerning attributes are shown in APPENDIX 10 and 11 respectively. In the website case actors just picked the most important attributes but could not order them. Therefore, in the calculations each picked attribute was simply given an average value 7. The much lower columns compared to domain evaluations indicate a much bigger variance between individual prioritizations. This is partly due to the big number of attributes and their overlapping meanings. With respect to attributes, different quality profiles of the two cases are in addition due to different attribute collections used in questionnaires.

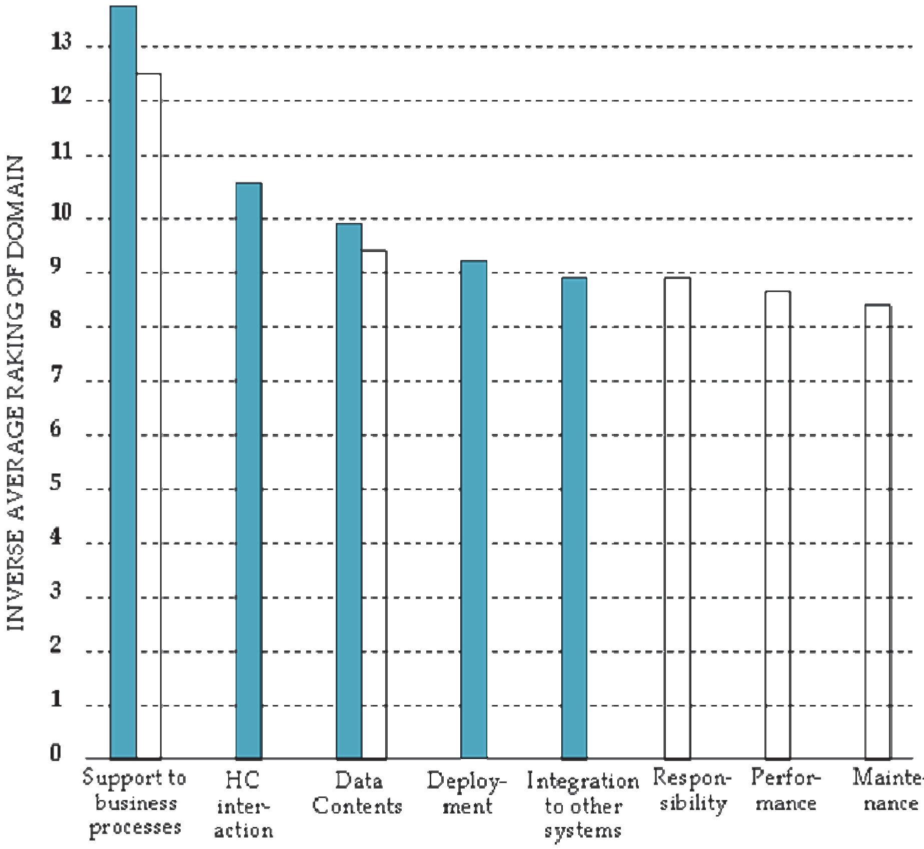


FIGURE 25 Top five EMIS and website domains

In connection with the EMIS case a synthesizing workshop was arranged on basis of questionnaire results to reconsider the top five qualities and check their order. As a result, only 'user friendliness', 'compatibility' and 'security' were kept among the top five attributes. The name "security" was changed to "safety" in order to avoid confusing this physical safety with authorization and privacy issues. Further, more weight was given to the prioritization of domains than attributes and accordingly 'usefulness' and 'coverage' were selected, instead of 'reliability' and 'installability', to represent two top areas of concern 'support to business processes' and 'data-contents' respectively. Although the domain-attribute collection on questionnaire disclosed neglected areas like 'authorization', 'architecture and design', 'maintenance', 'change' and 'performance', nothing was done to fill them.

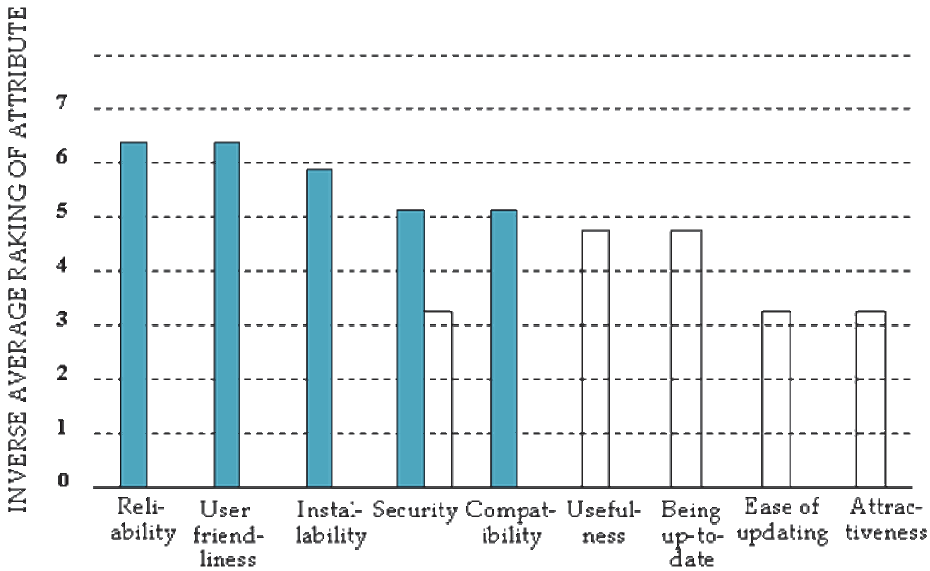


FIGURE 26 Top five EMIS and website quality attributes

The new top five attributes of EMIS were evaluated by workshop participants against each other and the evaluation quantified using a method called analytic hierarchy process (AHP) (presented e.g by Svahnberg et al. 2003 and Karlsson & Ryan 1997). In AHP all combinations of elements (in this case of the top five attributes) are evaluated pair-wise according to a certain scale. FIGURE 27 illustrates a questionnaire diagram for such a pair-wise comparison. It tells which of the two attributes is more important and how much more.

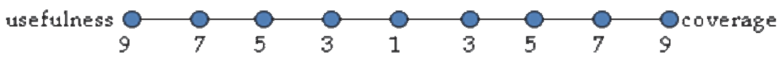


FIGURE 27 Diagram for pair-wise comparison of usefulness and coverage

Comparisons are then transferred into an *n by n* matrix (*n* is the number of elements). If, for example, usefulness is 7 times more important the value transferred to the matrix cell is 7/1 (=7). If in this comparison coverage is 5 times more important, the cell value would be 1/5 (=0.2). Thereafter an estimation of the eigenvalues for each of the matrix rows (in matrixes got from individual informants) is computed by a method called "averaging over normalized columns". The list of eigenvalues for matrix rows (corresponding to top five attributes) is called "priority vector". Finally the vectors from individual informants are synthesized into a combined view of all informants called "synthesized priority vector". This was done simply by taking the median values of all informants. The details of the process are described in Svahnberg et al. (2003). In the EMIS case, however, attribute pairs were compared only once and consequently there was no need to calculate a consistency ratio between two different comparisons (e.g. usefulness-coverage and coverage-usefulness) by the same informant. FIGURE 28 depicts the final synthesized priority vector of top five EMIS quality attributes: coverage (contents), usefulness, safety, user friendliness and compatibility (integration to other systems). Later on, after having the EMIS prototype at hand, user friendliness rose to the most critical attribute when entering the first raw data into database. If one compares the prioritizations against the description of Africa as environment for information systems in section 5.1, it is a little bit surprising that only safety is among the top quality attributes. One would expect also to see concerns like sustainability, maintainability and affordability.

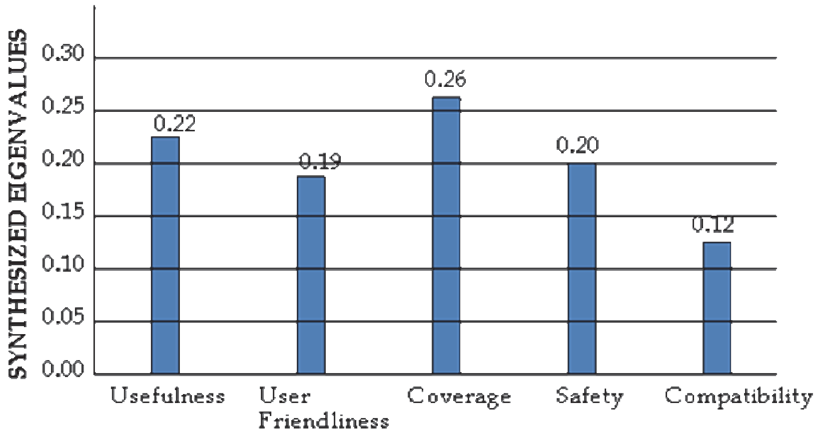


FIGURE 28 Synthesized priority vector of top five EMIS quality attributes

Prioritizing quality attributes is an essential part of general quality design. Letting actors identify priority areas is what also van Reijswoud (2009) found important in Tanzanian Health sector ICT projects. And in general, the idea of ranking qualities is comprehensible and sensible to informants. Both case studies showed, however, that *informants find it difficult to prioritize tens of attributes and the many domains included in attribute sets*. This is evidently due to several reasons, like the mere big amount of items (i.e. cognitive reasons mentioned by Miyoshi & Azuma 1993, 426), unfamiliarity with English language or ICT terminology, unfamiliarity with ICT as such, etc. In some cases this even led the informant to leave the questionnaire unanswered. Further, in very few system development projects the team has, after ranking the qualities, time to model in detail all the quality attributes that the informants have selected. This has been noted for example by Villalba et al. (2010, 32). A *substantial effect on product quality can be made*, however, *even by focusing on a small number of key attributes*.

The *best practice for prioritizing quality attributes is to prioritize first domains and thereafter individual attributes*. Both the domains and attributes must be given clear basic definitions. In addition, the set of domains and attributes should contain minimum amount of overlapping between elements. Using smaller subsets of domains and attributes (see discussion above in connection with attribute sets) instead of one all-inclusive set can also make the prioritization easier for informants. In the initial meta-model priority is depicted as an element of attribute. Prioritization is, however, always made in relation to other attributes. Therefore even in the meta-model a better *place for priorities is inside attribute set part*.

Attribution

In neither of the case studies informants were explicitly asked to define if the prioritized qualities are qualities of the whole system or rather of some of its constituents. Later the process of trying to find out the contributors constituents (see discussion below) led the researcher to consider these constituents, like user interface, as more precise owners of the qualities in question. The attribution elements were, however, in the end left unfilled in resulting attribute model prototypes. Consequently no definite conclusions can be drawn from the two first case studies concerning attribution element.

Use Case

According to the initial meta-model, creating attribute definition elements, called propositions and predicates, would have been the next step in quality design. However, it was found very difficult to elicit opinions from informants in a discussion based only on initial attribute definitions. *What helped was to think about how the information system is used as part of a particular business activity and what the required qualities would be at that point*. Therefore, some basic use cases were first singled out and attributes defined in connection with them. This process led also to *expanding the narrow definition of use case in the initial meta-model* and to introducing in addition the concept of 'scenario'.

In general terms "use case" and "scenario" refer to a particular activity involving the use of an information system, an activity performed by it, or an activity affecting it. *Business use cases*, as part of the business model, represent the processes of an organization with or without explicit reference to supporting information systems. A *system use case* (e.g. entering data), for its part, represents the use of system per se, without, in the first place, drawing attention to its connection to business processes. The use case can be seen from the point of view of either the user or the system itself. The latter way of thinking comes closest to the definition of use case in the initial meta-model, formulated by Jacobson et al. (1999, 35): "a sequence of actions that the system performs to offer some results of value to an actor". Finally, the term *scenario* refers to particular circumstances, or to a flow of events where the role of human actors as users of an information system is non-existent, or not focused on. Kazman, Bass, Klein, Lattanze & Northrop (2005) use scenario as an overall notion and wrap use cases under it. They decompose use cases into six parts: stimulus, response, source of stimulus, environment, artefact stimulated, and response measure. In the case studies this was found to be too complex. Instead, brief or initial step-by-step use case descriptions were used (cf. Jacobson et al. 1999, 149-150). Sometimes, however, a note was made of what instigates a use case or a scenario, and under what conditions such an event occurs. FIGURE 29 depicts the first business (upper part) and system (lower part) use cases identified within the Ministry's statistical unit by the project team. O1 and O2 stand for organization 1 and organization 2 respectively. Numbers inside ellipses were used to identify use cases and scenarios in documentation.

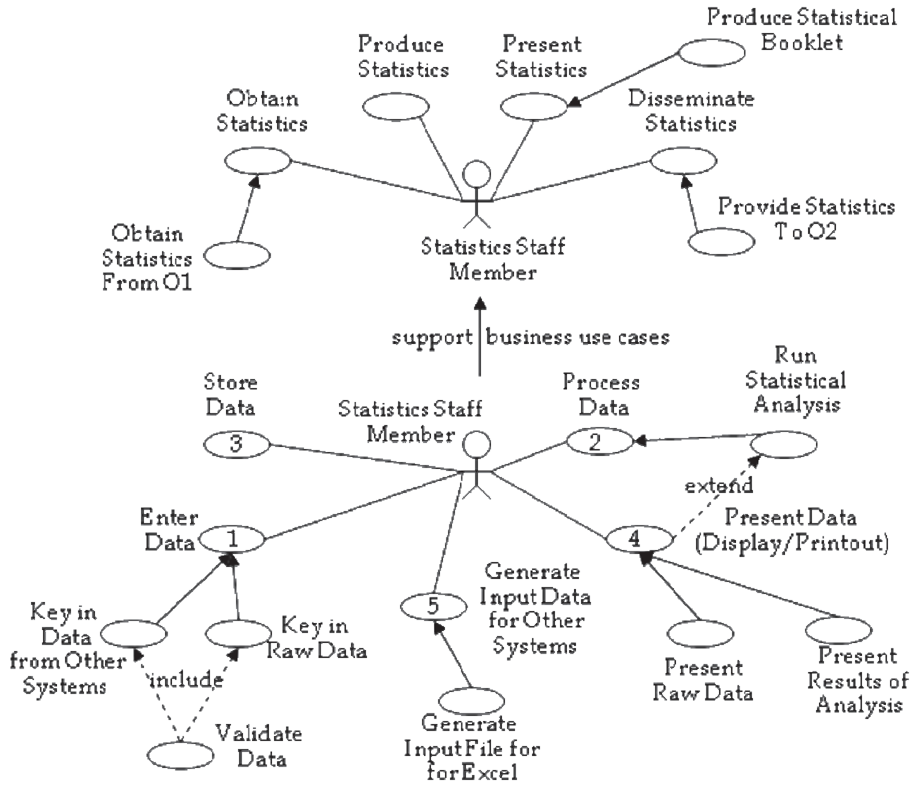


FIGURE 29 Basic EMIS use cases

Four of the top five quality attributes (coverage, usefulness, user friendliness, and compatibility) were considered in connection with the following system use cases: “Enter Data”, “Process Data” and “Generate Input Data for Other Systems”. Events or processes identified as dealing with safety were categorized as scenarios. FIGURE 30 depicts some scenarios related to the safety of the MIS. The diagram is a high-level description, in which the scenario names refer to the consequences of event flows. Damage to cable refer to African environmental conditions, where small animals like rats can cause problems.

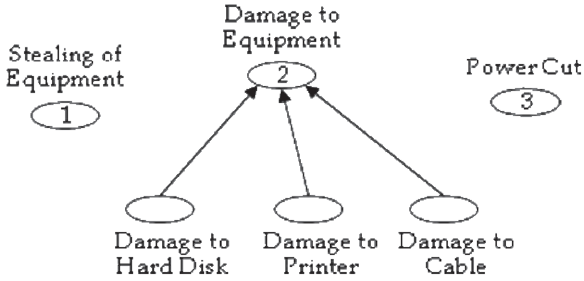


FIGURE 30 A selection of scenarios related to safety of EMIS

FIGURE 31 depicts the first use cases identified and used in the website quality modelling. They reflect the needs of Maputo program office staff and show a quite different context of use compared to the EMIS.

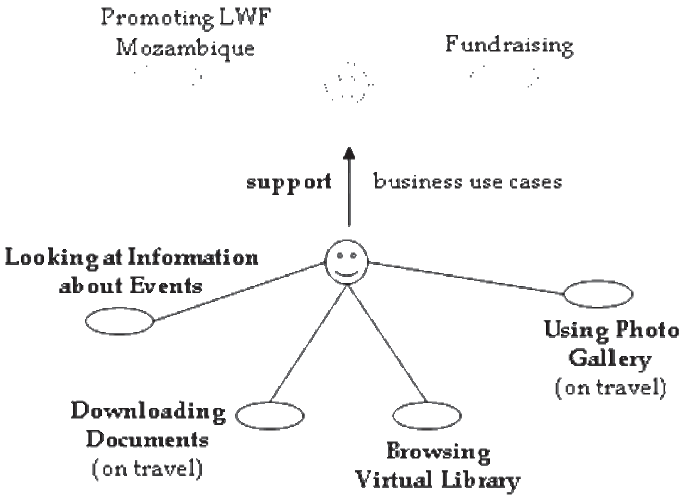


FIGURE 31 First use cases identified in website case

Definition

Work on this model element started in both case studies with writing down informal notes on discussions about what it means, and what would indicate that the planned system (or a part of it) has a particular quality in connection with certain use case or scenario. These notes can be called pieces of discourse, suggestions or assertions. The notes were then transformed into formally correct sentences, and finally into a collection of propositions (or statements) and predicates. Since for the most part they are expressed in ordinary language, they can be regarded as part of the "shared" language between different actors.

TABLE 10 and 11 show respectively sample propositions and predicates belonging to attributes coverage and usefulness in EMIS case. TABLE 12, in turn, shows some simple definition elements of usefulness in website case. Predicates were not generalized into attribute models in the latter case. These kinds of tables encompass all the definition elements of a given quality attribute. The first propositions, sifted out from discourse, are often simple and self-evident, but in the course of the work they soon become more numerous and advanced. Both case studies highlighted the fact that often the individual propositions are valid only in connection with specific use cases or scenarios.

TABLE 10 Sample definition elements of coverage

Domain	Data.	
Dictionary-Type Definition	The set of information or categories of information contained in MIS is sufficient.	
Use case	**Proposition**	**Predicate**
Enter Data	Statisticians are able to enter all the statistics from standard data collection forms into the MIS database.	Group G of users are able to enter a specific collection of data (CD) into a database (DB).

Statements in one form or another have been an important element in scientific writings on quality. Statements, as they are understood in this study, correspond, for example, more or less to the response and response measure elements of the scenario-based analysis used by Kazman et al. (2005). They have also similarities with how requirements are expressed when related to the ISO 9126 attributes by Egyed & Grünbacher (2004). Cysneiros & Leite (2004), for their part, use the following notions taken from Chung, Nixon & Mylopoulos (1999): 'goal', 'subgoal' and 'operationalization'. Operationalizations are the most specific or leaf-level propositions in their goal hierarchy.

TABLE 11 Sample definition elements of usefulness

Domain	Support to business processes of the user organization.	
Dictionary-Type Definition	You can use MIS to do something. MIS can help you to achieve something in a particular situation.	
Use case	**Proposition**	**Predicate**
Process Data	Statisticians can make all necessary analyses using the system.	Group G of users can carry out a specific group of tasks (GT) using system S.
Process Data	Statistician can produce all necessary statistical presentations using the system.	Group G of users can carry out a specific group of tasks (GT) using system S.

TABLE 12 Sample definition elements of usefulness (website case)

Domain	Support to business processes of user organization.	
Dictionary-Type Definition	Website can be used to do something. Website can help to achieve something in a particular situation	
Use case	**Proposition**	**Predicate**
Looking at information about events on travel	Staff members can follow what is happening at program and project levels. In addition they check deadlines for reports.	
Looking at information about events on travel	Staff members can harmonize personal calendars with the common project calendar.	

In addition to grouping statements by use case, as is done in the tables above, they can be categorized in other ways too. Cysneiros & Leite (2004) differentiate behavioural statements (behavioural responses) from (in their terms) notions that express the meaning of a symbol. They also divide operationalizations into dynamic and static. The latter are supported with some data store in the system, while the former call for some actions to be performed. Accordingly, TABLE 10 above (for example) contains a static statement and TABLE 11 contains dynamic statements. Developing in advance *a general scheme for categorizing statements would greatly help informants*.

Both case studies showed that *categorized proposition tables* are a *satisfactory* and quite comprehensive *way of defining the quality attributes*. They form also a good *basis for designing a quality procedure*. Predicates represent a level of generalization out of system specific propositions. Actors might not, however, be familiar with academic concepts like 'predicate'. Developing a complete set of statements and propositions for all attributes is a long process and may continue even during the quality measurement stage.

Related to

Analysis of the *definition sentences* and propositions of quality attributes shows that they *describe in fact how the information system* or its constituent that "owns" the attribute *should according to the informants be related to the environment*. It is the "subject matter" of the propositions. This seems in general to be the essence of quality. Qualities equal in the end to these relationships or manifest themselves in them. The relationships can be described on a more abstract level by using verb phrases that in logic are called polyadic predicates as was done in tables above.

All of the top five attributes of the EMIS case were modelled in terms of relationships between the system and elements in its environment. Usefulness is a tool-actor-activity (or resource-actor-activity) relationship and relates the EMIS or its parts to particular business use cases or larger processes like poverty monitoring, sectoral reform efforts, programme implementation, decision making, managing sectoral development, reporting, processing and disseminating data, solving problems and gaining knowledge. User friendliness relates the users and EMIS to each other. Compatibility relates EMIS to other information systems and coverage to the datasets that must be covered for particular reasons. Finally, safety propositions express how the system reacts to external events like power irregularities. Similarly, the two top website attributes that were modelled together with actors describe the system-environment relationships. Usefulness relates the website to program and project activities, staff's personal calendars etc. Being up-to-date means that website contents corresponds to current circumstances. The remaining three attributes, ease of updating, attractiveness and security can as well be described in terms of system-environment relationships.

Contributors

Analysis of quality attribute definitions and first sketches of attribute models led to the introduction of a new attribute meta-model element. *None of the initial meta-model elements could explain how to realize the qualities in practice*. Obviously there must be something at least in the system itself that plays this part. Accordingly those *system constituents and structures that directly or indirectly affect the system-environment relationships* discussed in previous section *were named as contributors*. This was actually implied in the initial meta-model and theoretical background but went unnoticed at the beginning of the study. It is very clearly stated by Dromey (1996). Even McCall's (1977, section 4, page 4) definition of accuracy as "those attributes of the software that provide the required precision in calculations and outputs" points to the same direction as well as functional requirements in Boehm et al. (1978). The initial meta-model looked at the relationships between the information system and its context and the members of these relationships, but did not account for what affects the relationship and its intensity. Case studies in real environment demanded the quality modelling team to give programmers ideas about how to help for their part in realizing the quality expectations. By combining this observation to what was stated in connection with 'related to' element above it can be concluded that the essence of quality to a particular actor or set of actors can be basically described with two core elements of attribute model: 1) desired relationships between the information system and its environment and 2) contributors.

It can be easily understood that the data model contributes greatly to the coverage, as defined in connection with the "enter data" system use case of the EMIS. In addition, user interface objects must support the entry of all data items. Similarly, the system behaviour embedded in methods, classes and packages determines for its part the usefulness of EMIS. Furthermore, certain user interface objects contribute to user friendliness, certain features of data-model and system interface to compatibility, and finally a number of hardware related elements to safety. Similarly initial contributors were identified in the website case. Different software modules, like "document manager" and "events", contributed to usefulness, administration interface to being-up-to-date and so on.

Cysneiros & Leite (2004) present a technique for relating quality attributes (non-functional requirements) to use case, class, sequence, and collaboration diagrams. This technique is a means to indicate exactly which design elements are responsible for satisfying individual quality requirements. This kind of detailed analysis could not be used in connection with the EMIS case study due to the lack of design documents. Despite repeated requests made by the researcher, these documents were never made available. Either the documents did not exist, or else they were kept secret for business reasons. Reverse-engineering, for its part, would have been too cumbersome.

Connected to

In the initial quality meta-model, connected to -element belongs to the attribute part of the model and is positioned right after related to -element. However, because the connections are relationships between attributes, it was felt in the course of case studies that *the place for this element is actually in the quality attribute set*. Further it was seen that *full understanding of attribute connections is achieved only after their values have been measured*. This holds true especially with conflicts. Nevertheless, the relationships can be predicted before that by comparing attribute definitions and contributors.

In the EMIS case, modelling and measuring quality attributes did not proceed far enough for to make precise account of attribute interrelationships. In general terms, however, coverage, compatibility and even user friendliness was viewed as supporting usefulness. If, for example, some vital data is missing, this will hamper the usefulness of the system. On the other hand, there is obviously no conflict between the top five qualities. The only attribute conflict noticed was that between user friendliness and simplicity of design. An interface that was easier to use, tabular and form-like for the purpose of entering raw data would require more complex design objects on the underlying architectural layers. FIGURE 32 shows a way to depict the relationships of top five attributes together with priorities in EMIS case. The arrow above the matrix shows the direction of relationship and numbers equal to attribute numbers in the pie chart. The angle of the slices represents the priority value (as a number in brackets) of attribute. Different symbols can be used in table cells to indicate the type of relationship. "+" sign stands for a positive contribution. In the website case attribute connections were not modelled.

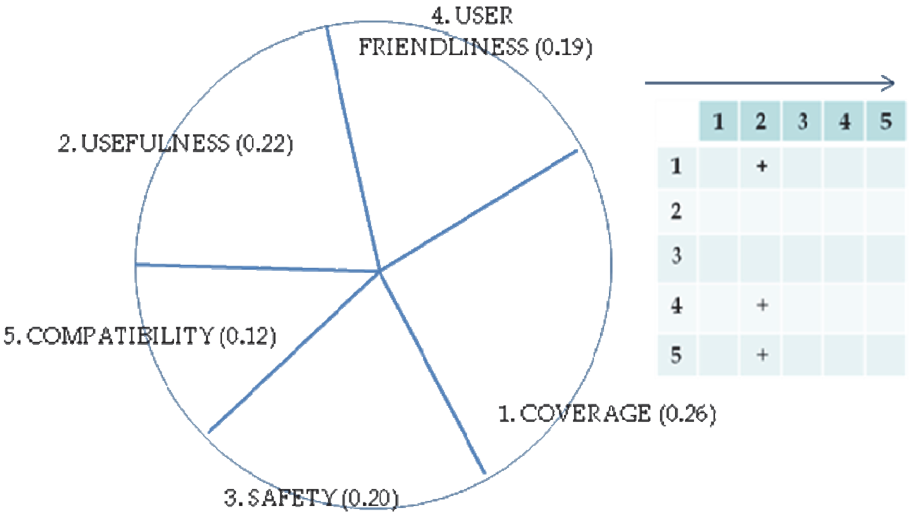


FIGURE 32 Priorities and positive contribution between top five EMIS attributes

Measurement, value and relative to

It was stated in the introduction and research design chapters of this study that metrics is not a primary part of the proposed quality meta-model. The ideas for measurement grow, however, naturally out of attribute models and the actual quality of information systems is of great interest to actors. Therefore, in the case studies the design of metrics was started and initial measurements made whenever possible. The *definition elements of each attribute model* created in the course of detailed quality design *formed a basis for working out how to measure the existence and degree of the quality in question*. Some questions used in eliciting propositions like “How would you check that the system has the quality Q?” pointed directly to measurement. The team was aware that a complete measuring arrangement would include careful selection of the instrument, unit, scale, actors, and measurement procedure. In addition quite ambitious goal values were set for some attributes.

After modelling the top five EMIS attributes by instantiating the meta-model elements from definition and attribution to contributors and metrics each of them were given a general graphical presentation. FIGURE 33 exemplifies the presentations by depicting some of the main features of ‘coverage’ in connection with ‘process data’ use case.

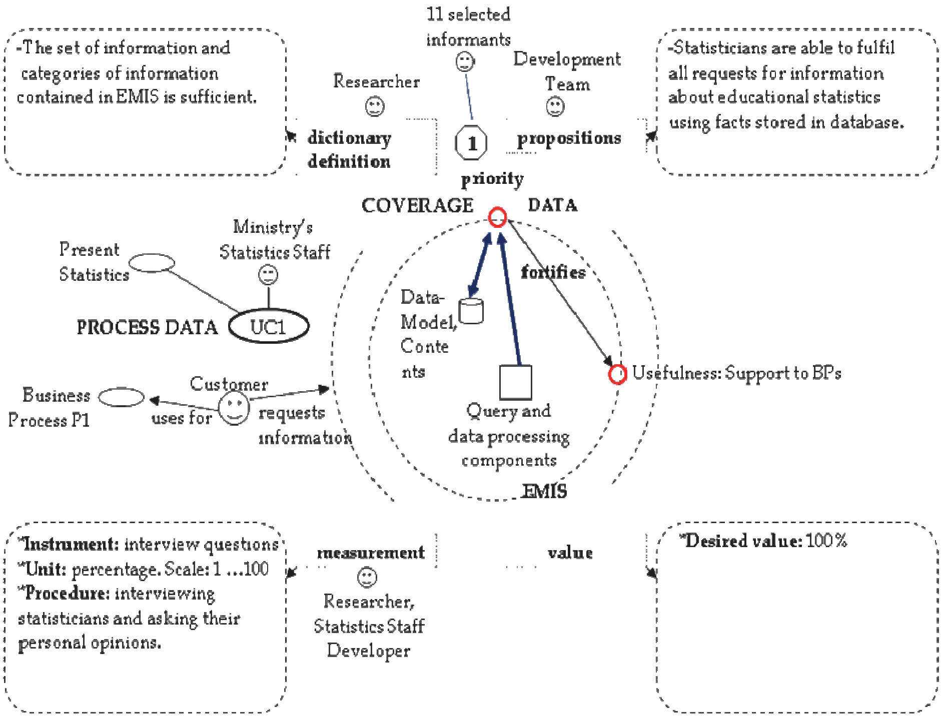


FIGURE 33 Simplified graphical presentation of 'coverage' attribute model

The model in the figure is simplified. It shows only one sample proposition, some of the contributors, one connected attribute, etc. In general it is often impossible to fit all information to one diagram. This problem has to be solved by representing only core features graphically and attaching textual descriptions or distributing the information on several diagrams. The human actors in FIGURE 33 indicate persons or groups who are responsible for definitions, prioritization and measurements or have a role in the use cases symbolized by ellipses.

The first measurement of EMIS top five quality attribute values was conducted in May–June 2006. In that stage the first statistical data was being entered into the system. Therefore only user-friendliness, coverage and safety were measured and only in connection with enter data use case. Actors did not have enough experience of system's usefulness or compatibility to other systems yet. Measurements were based on 17 propositions. Majority of them, 15 statements, were related to user friendliness, one to coverage and one to safety. The researcher conducted a semi-structured interview in which four statisticians plus one developer was asked (with respect to each proposition) whether they thought that the first version of EMIS had the feature in question or not. Possible answers were "yes", "no" and "cannot decide". This procedure resembles the marking of sub-goals as satisfied, partly satisfied or denied in Cysneiros & Leite (2004). The assessments were afterwards quantified, so that if *yes* answers were more than *no*, it was given a score of 2. A minority yes-response, in turn, was scored as 0, and a "tie" scored as 1. In this arrangement, the measuring instrument was a set of questions, and the unit a natural number on a scale from 0 to 30 (for user friendliness) and from 0 to 2 (for coverage and safety). The actors needed for the measurement were the researcher and the interviewees, and the procedure consisted of semi-structured interviews combined with a simple quantification method. *The measurement process in itself was experienced as very useful*. It *produced refinements and additions to the proposition sets, and* at the same time *to the measuring instruments*. The interviews also disclosed gaps in some users' knowledge about how to use the system most effectively.

A second limited control measurement was made in June 2007. In this interview propositions were not used anymore. Actors were asked only to evaluate in percents how far from maximum the value of each top five attributes is currently. This time a rough idea was got also about the compatibility and usefulness of the new system. FIGURE 34 shows the measured quality of the EMIS with respect to all of the top five qualities in 2006 and 2007. The highest scores were achieved for data coverage and the safety of the system. The variation (marked with horizontal divider) in the column representing data coverage in 2007 relates to different results in connection with two different use cases. From the viewpoint of entering raw data, the coverage has grown from 95% in 2006 to 99–100% in 2007 (second column). However, when measured in connection with processing statistical data, the experienced coverage was still only 70%. This means that the database did not contain enough information to create all the required statistics, although the data model would have permitted entry of the information. It was also understood that the demand for new kinds of statistics is continuous. The observed safety of the system was complete at the time of the case study. Even when a voltage regulator burned out in 2007, no harm was caused to the EMIS.

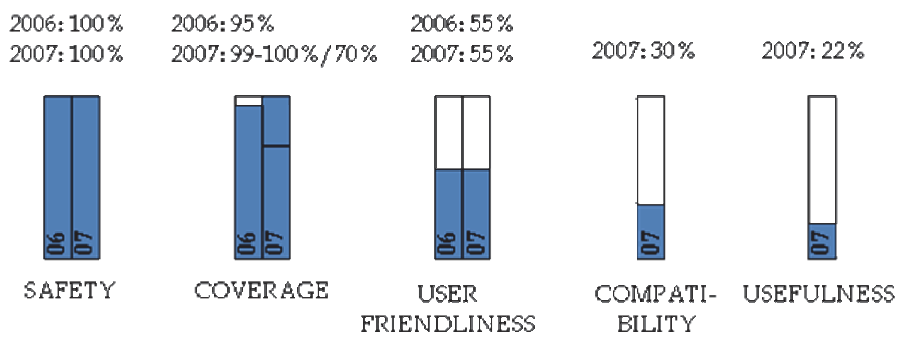
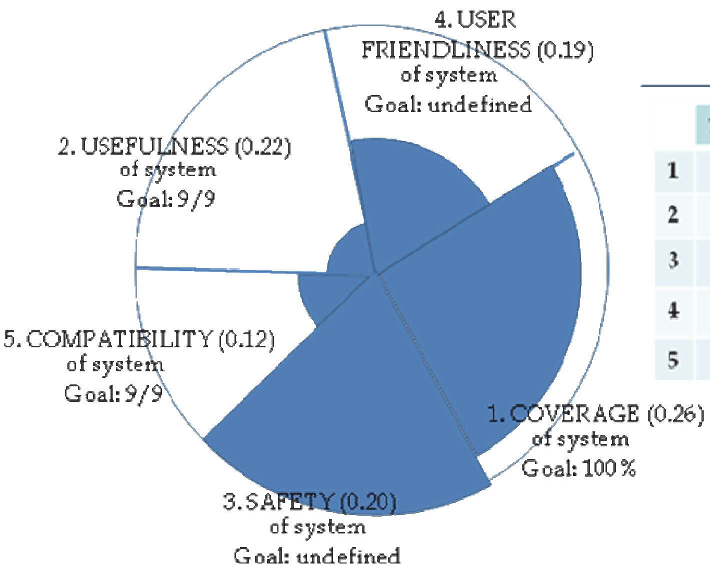


FIGURE 34 Measured values of top five EMIS attributes in 2006 and 2007

The measured user friendliness of the first program version with respect to the use case "enter data" was 55% of the maximum in 2006. However, no change took place between the two measurements, but it was felt that the time needed for entering data had been somewhat reduced because of growing familiarity with keying in data. The lowest scores were measured for usefulness (2/9) and compatibility (3/9). In the view of the statisticians, the system was still only a data store and a replacement for the old COBOL application. Only a few printouts had been given from the system to customers who required details not already shown in the traditional statistical booklets. Regarding compatibility, expectations that the new EMIS could produce statistical analyses or at least create fully compatible output files for Excel were not yet met. Similarly the system did not produce any outputs directly for stakeholder's information systems. Another way to visualize the current fulfilment of quality requirements is shown in FIGURE 35. The angle of the slices represents the priority value of attribute and the radius of filling the achieved value at the time of measurement in June 2007. A similar technique is used for example by Miyoshi & Azuma (1993). This kind of figure gives the widest view of overall quality design and its realization. It shows priorities, attributions (in this sample case all to the system as whole), goal values and attribute relationships.



	1	2	3	4	5
1		+			
2					
3					
4		+			
5		+			

FIGURE 35 Fulfilment of EMIS quality requirements in June 2007

Cysneiros & Leite (2004) argue that *quality requirements can rarely be fully satisfied*. Hence, (following Herbert Simon 1996) they use the verb "satisfice" to underline the notion of partial satisfaction (cf. also Chung et al. 1999, 8). The EMIS *case study supports this view*. And despite the ambitious goal values given to some attributes, it was clearly understood within the project team that most of the quality definitions for example aimed only at agreement on what could be a sufficient value for each attribute. In addition to the partial fulfilment of quality requirements, the *values measured for individual quality attributes are often relative in certain respects*. This has been widely noted in articles concerning quality (e.g. Berki et al. 2004, Voas 2004). In other words, one and the same information system may have, not have, or have to a variable degree, a particular quality when measured, for example, in connection with different use cases. Usefulness is relative to the business processes and tasks for which the system is used, coverage to the datasets that must be covered, and compatibility to the information systems connected. Even safety can be viewed as relative to the threats – for example power outages, sags and surges – against which the system must be protected. The same relativity holds true also with respect to the prioritization of attributes carried out in the course of general quality design. In connection with different use cases, scenarios, data sets etc., the order of priority within the very same set of attributes can be different.

The *individual qualities of an information system* or its constituents *are defined, prioritized, observed and measured in a frame of reference* that is in some ways similar to that in physics, where there is a set of axes relative to which an observer can measure the position and motion of points in a system. A classical and simple example is that of two people standing on either side of a street and watching a car driving past. For one person the car is moving towards the right, for the other it is moving towards the left. The observers of an information system quality are all the actors and informants around the system. For one who measures a quality, the things that cause relativity constitute together the set of axes or the frame of reference relative to which measurements are made. They include particular business and system use cases, business processes, observer's organization and role in it etc. As noted above in connection with actor element, several different groups of stakeholders can be listed in connection with EMIS. However, only statisticians at the Ministry took part in the measurement of quality attributes during the case study. Other groups obviously had viewed the system from within a different frame of reference. Coverage and usefulness, for example, will undoubtedly mean different things to district-level officials than they will to employees of the National Bureau of Statistics. Both groups of stakeholders are interested in different data sets, and they will use the EMIS to support different business processes. In addition, these kinds of factors will change from time to time.

Reflections on EMIS *case point clearly out the relativity of quality attribute definitions and measured values*, as well as the difference between actual and desired values. The relative nature of several quality model elements suggests *removing relativity from the meta-model as an individual element*. Further, the case study made clear the importance of proper measuring arrangement. Measuring quality must be carried out in a controlled manner (the instrument, unit, scale, actors, and measurement procedure). A separate metrics and measurement process meta-model could give guidance in that respect.

5.2.4 Concept of quality

In the theoretical background extensive and explicit definitions of the concept of quality were rare. ISO 9126 (2001) adopts the definition of quality given in ISO 8402 (1994). According to it quality is "the totality of characteristics of an entity that bear on its ability to satisfy stated and implied needs" (ISO 9126 (2001, 20)). Compared to findings of case studies, ISO definition points clearly more to something (totality of characteristics) inside the system than to the relationship between system and context. The latter is better a result of the quality than the quality itself. The initial meta-model did not define the concepts of 'information system quality' and 'quality requirement' either. According to the research design understanding of these concepts should grow out of the case studies and accompanying conceptual analysis. The most effective approach is to look at the individual attribute models.

In the initial meta-model a quality attribute was simply defined as "a good characteristic of an information system" that is closely related to the satisfaction or dissatisfaction of actors around it. These kind of characteristics are usually given names using abstract nouns derived from adjectives (the "-ilities") and traditional quality models present categorized lists of them. Syntactically an adjective is simply a unit that modifies a noun. Semantically it can, however, refer to different things and not only to some internal and inseparable features of an entity denoted by the noun. In the course of the two case studies a combination of model elements were used to look at selected quality attributes from different angles and to specify in detail each of them. This exercise turned repeatedly attention to the relationships between information system and its context. *Careful analysis of propositions, for example, discloses that most of them in the end point to or even explicitly specify a relationship between the system and its environment*. Usefulness (one of the top five attributes in EMIS case study), as a typical example, was found to refer to a relationship between the physical EMIS (object), statisticians and a particular business task which, in turn, is part of some larger business process.

The *new contributor element*, in turn, *suggests asking with respect to each structural-behavioural element* (see discussion above in section 5.2.3), *to what does it contribute*, why is it required? Some constituents may be needed to implement other constituents, but *sooner or later the chain will lead out of the internal view to some required relationship between the system and environment* that justifies the existence of the internal elements. This observation suggests that defining and estimating information system quality in general should also start from this angle. In other words, the *essence of quality experience is in the end fulfilment of expectations about what happens between the system and user or some other elements in the environment*.

Total or overall quality of an information system is often differentiated from individual qualities. According to proposed meta-model and the analysis above, *calculation or assessment of overall quality must be based on the degree of fulfilment of overall quality design that is embodied in the system specific attribute set* which defines the priorities and goal values, and describes the interrelationships of quality attributes. If any mathematical or other formula is created to derive the value of overall quality out of measured values of individual qualities, it must take into account the priorities by giving different weight to values according to them.

5.3 Intermediate Meta-Model

FIGURE 36 depicts an intermediate version of meta-model after first two case studies. It visualizes the implications of the studies for the meta-model's elements and structure. The main points and modifications were:

- Overlapping rectangles symbolizing the model element relationships were omitted in order to see if a less complicated diagram is more understandable to its users.
- Actor element is positioned outside and above other elements to symbolize the importance of actors in implementing any other model element.
- Prioritization is an element of attribute set, even if individual attributes can be attached a ranking number.
- Use cases can be differentiated from scenarios.
- Use case and scenario element is positioned above definition element to underline that they can be defined before the latter and used to boost eliciting definition statements. An inner rounded rectangle groups together the core model elements with respect to individual attributes.
- A better name for 'related to' is relationship. It refers to certain relationship between system and its environment, the subject matter of definition sentences.
- Contributor is an important new meta-model element and one of the core aspects of quality.
- Like priority, 'connected to' belongs logically to attribute set. Connections are connections between attributes in the set. Different general goal values can be attached to some attributes in the set to indicate, for example, intended balance between conflicting attributes.
- 'Relative to' element is removed from the model and value element positioned into metrics part. Relativity is inherent in quality and is encountered in connection with many elements. It can be pointed out in the general description of quality meta-model.
- Due to the importance of proper measurement procedure (even if it does not belong to the actual quality model) metrics part has been expanded to cover instrument, unit, scale, value and procedure elements.

As a result of the above considerations the meta-model was presented to actors in a diagrammatical shape that can be seen in FIGURE 36. This happened until the diagram got its final form at the end of the third case study.

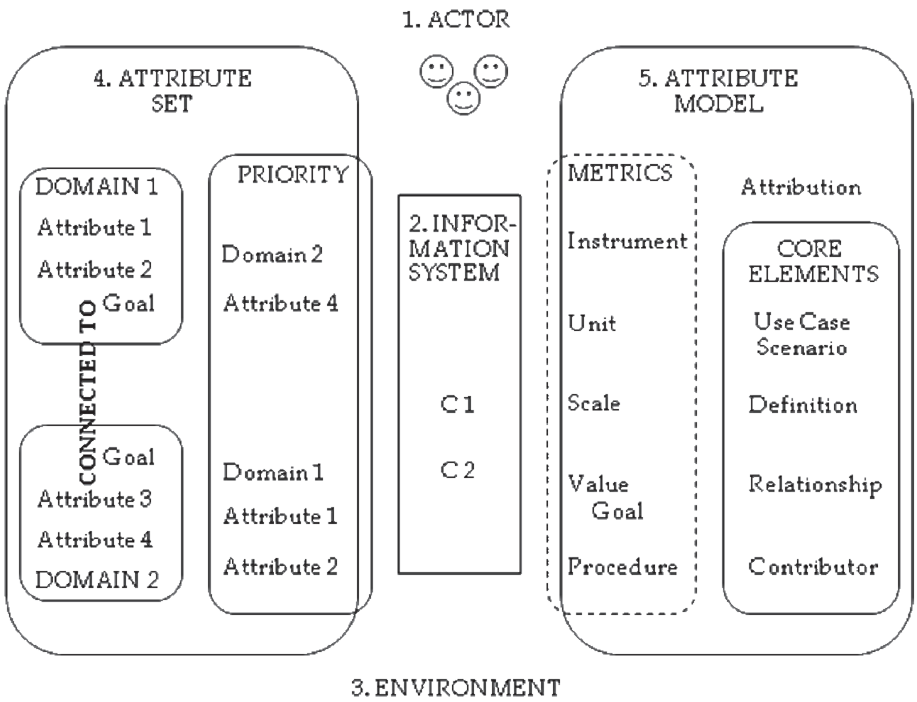


FIGURE 36 Intermediate version of quality meta-model

5.4 A third case study: land registration system

5.4.1 The Case

The third quality modelling case study was conducted in connection with development of a Land Registration System (LRS). LRS is a component of Zanzibar Land Information System (ZALIS) which is intended to be a wider system of systems for managing different kinds of land information. The system development activities were carried out as part of a development co-operation project funded by Finnish Ministry of Foreign Affairs and Revolutionary Government of Zanzibar. The project is known by name Sustainable Management of Land and Environment (SMOLE). The first phase of SMOLE was implemented 2005-2009 and the second phase started 2010 and continues up to the end of 2013. The main purpose is to provide technical assistance and initial investments for ensuring that sustainable land and environmental management practices are used in Zanzibar. Beneficiaries of SMOLE project are different departments of two ministries: the Ministry of Water Construction Energy and Lands and the Ministry of Agriculture Livestock and Environment. The first version of LRS serves especially two of the departments: the Department of Survey and Urban Planning (DoSUP) and the Department of Lands and Registration (DoLR). The organization structures of these departments are depicted in appendixes 3 and 4 respectively.

The development of LRS started in 2009 along with a building inventory made in Stone Town that is a kind of "capital" of Zanzibar. The town was included in UNESCO's World Heritage Sites in 2000. FIGURE 37 shows a typical view in the town. The survey gathered data about the use of buildings and occupation units, their size, public utilities, etc. Also photos were taken and later the building footprints digitized. This anticipated the actual adjudication and land registration. In an ancient densely built town the plots usually coincide with the building footprints. In connection with building survey the occupants were given forms that they could fill and use in claiming the ownership of the land on which the house was built. The claim information was also entered into the database. In 2010 the LRS was finally expanded with components for managing the parcels and property rights, including land leases. The quality modelling case study was carried out parallel to this development phase. Data about plots and structures inside them is intrinsically spatial. Accordingly besides traditional presentation forms, like tables, different kinds of maps are characteristic outputs from the LRS. Intended key consumers of the building, land and ownership information are Zanzibar Municipal Council (ZMC) and Zanzibar Revenue Board (ZRB).

All system developers have been up to 2011 foreign consultants, the researcher as designer-programmer, assisted by GIS specialists. Chief technical advisors and land registration advisers of the SMOLE project have also given their important contribution. Along the process the project has been looking for local ICT people inside the departments to be trained in system administration and system development. This did not work out very well. Therefore a decision was made in the end of 2010 to hire four Tanzanian professionals outside the ministry in order to form a transitional ICT unit that could in the course of next two years gain the knowledge of managing LRS and developing it further. At the time of writing the thesis the unit is established and working.

FIGURE 37 A typical Zanzibar house and narrow street in Stone Town

5.4.2 Workflows and steps

The quality modelling process in the third case study was in respect to the workflows identical to the previous ones. The steps listed below and their order, however, were different (major differences indicated in brackets). The steps equal with meta-model elements and numbering corresponds to numbers in FIGURE 38.

1. Identifying stakeholders, their perspectives and views, and selecting informants.
2. Building overall understanding of system and its environment.
3. Creating an overall quality design for the system, starting by selecting and prioritizing areas of concern (domains) using a questionnaire.
4. Identifying use cases and scenarios relevant to prioritized domains using a second questionnaire. (Taking this step right after prioritizing domains is different compared to previous cases.)
5. Listing positive and negative facts that would indicate existence or lack of quality in connection with each use case and scenario. (Dealing with definition elements already in this stage, before even naming attributes, is different compared to previous cases.)

6. Identifying quality attributes (attribute names) that refer to the positive facts listed in previous step and prioritizing them.
7. Attributing individual qualities to lower level system constituents if relevant.
8. Identifying contributors.
9. Modelling the relationships between the system and its environment that pertain to the individual qualities. (Delaying relationship analysis to this point is different compared to previous cases.)
10. Identifying connections between attributes.
11. Discussing initially about metrics.

The third case study paid more attention to actor perspectives and view than the two first ones. Modelling overall quality started by only prioritizing domains. Dealing with quality attributes (identifying, prioritizing, attribution to system constituents) came first after listing use cases relevant to each domain and facts indicating high system quality with respect to these use cases. Positive facts correspond to the concepts of discourse and propositions (step 5) in previous case studies. Finally, identifying contributors happened before definition of system-context relationships. FIGURE 38 depicts graphically the quality modelling process carried out in third case study.

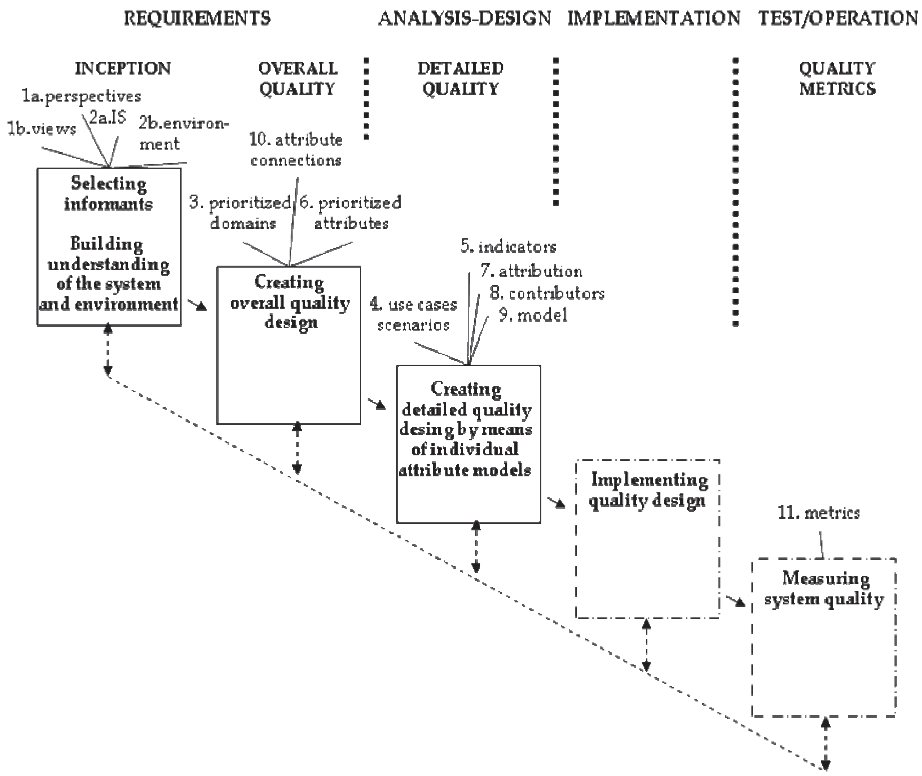


FIGURE 38 Quality modelling activities and system development workflows in third case study

5.4.3 Model elements

The third case study was based on the two previous ones. The meta-model in use was the intermediate version presented in section 5.3 and all the findings of two first cases were in the mind of the researcher.

Actor

As with previous case studies, it was again experienced that *listing all relevant stakeholders is tedious and building and managing a complete project group in that respect would be practically impossible*. It was felt that it is more feasible to start with most immediate system users and a number of other easily available stakeholders, and expand the number of informants afterwards according to needs. Therefore, in the first phase a group with 15 members was formed to carry out selection of most important areas of concern. It included employees from land registrar's office, ZALIS office and some foreign advisors. Group members and their roles are listed below in connection with perspective and view. Again the local actors represented the minority of population having experience of computers and Internet. The list includes in addition two actors, regional ICT advisor and head of ZALIS, who did not take part into first steps of quality modelling. In the second phase, modelling individual qualities, took part only four actors: international GIS advisor, international land registration adviser, regional ICT advisor and the local assistant registrar. Some of the actors are seen in FIGURE 39 that was taken during the quality model evaluation workshop. Like in pervious case studies, the researcher tried to act as an editor of the quality model and not to influence the selection and definition of quality requirements.

FIGURE 39 LRS quality model evaluation workshop participants

Perspective and view

In previous case studies it was clearly understood that different actor views have an impact on the resulting quality models. This understanding was reinforced during LRS case study and conceptualized as notions of perspective and view. *Each actor has a particular perspective on and view of the information system* under development that is determined by actor's knowledge about information systems in general, work, roles in organizations, and experiences with this particular information system. In LRS case there where different opinions even about such basic things as which organization should be hosting, developing and administering the system. In addition to actor or actor group specific views, there are *shared views* of the system. These are those published system and environment descriptions that are presented and made available to all actors.

The importance of *views and perspectives* suggests that these notions *must be reflected in the quality meta-model as elements* together with actor. FIGURE 40 depicts the different perspectives and views effective in land registration system quality modelling. They comprised four different organizations: SMOLE donor project (Ministry for Foreign Affairs of Finland (MFAF)), and three departments of Ministry of Water, Construction, Energy and Lands (MWCEL) of Zanzibar. The departments were: Department of Survey and Urban Planning (DoSUP), Department of Land Registration (DoLR) and Department of Land Administration (DoLA). Individual actor's view usually comprises only certain parts of the environment and certain parts of the system. These views can be broadly categorized into external and internal views, the former seen by managers and end users and the latter by developers and database administrators. In FIGURE 40 solid line connecting the perspective and system indicates that the actor had an internal view of the system as a developer, and dashed line refers to the first end users with external view.

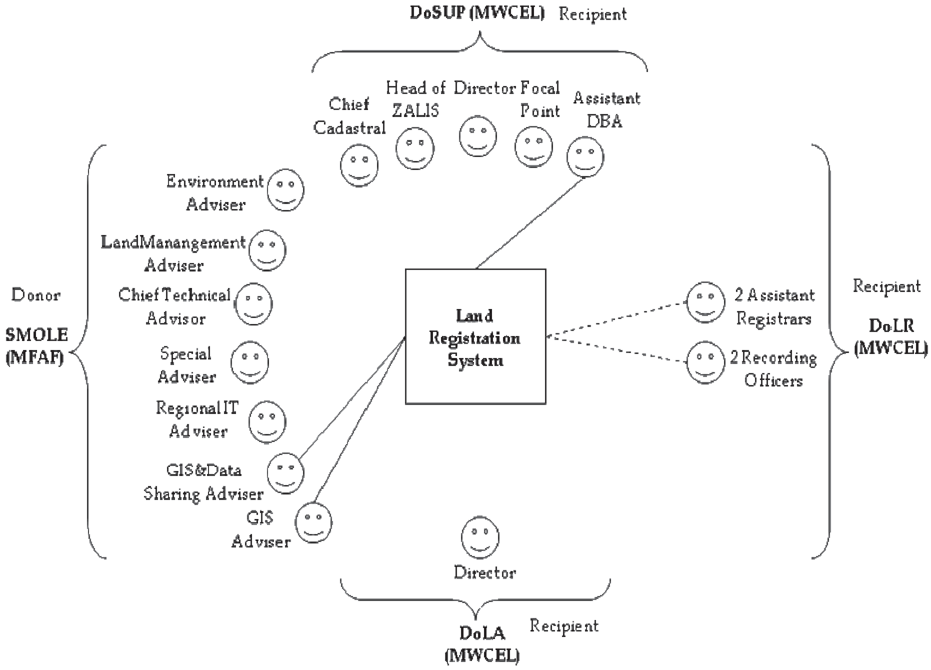


FIGURE 40 Actor perspectives in LRS quality modelling

Descriptions of information system and environment below give some additional information about the initial shared views that were held in connection with MPC quality modelling.

Information System and Environment

Like in previous case studies the general African conditions described in section 5.1 were well know to actors. In other respects the information system and its environment were initially described using simple diagrams that were attached to the questionnaire dealing with prioritization of domains and attributes. Accordingly the informants in this case study had a bit better initial understanding of the target system and its environment than in previous studies. The predominant view was, however, again the black box view. One of the diagrams (FIGURE 41) presented the system as a "black box" surrounded by names of the most important user/stakeholder organizations. ZRB stands for Zanzibar Revenue Board, ZMC for Zanzibar Municipal Council, MALE for Ministry of Agriculture, Livestock and Environment. Halmashauri is the Swahili name for other municipalities. Agriculture and Environment refer to departments of MALE. ZICRO is the acronym for Zanzibar Identity Card Registration Office that hosts a kind of population registry. Comparison to FIGURE 40 shows that only three of the stakeholders were represented in the quality modelling actor group, which poses a risk to quality design.

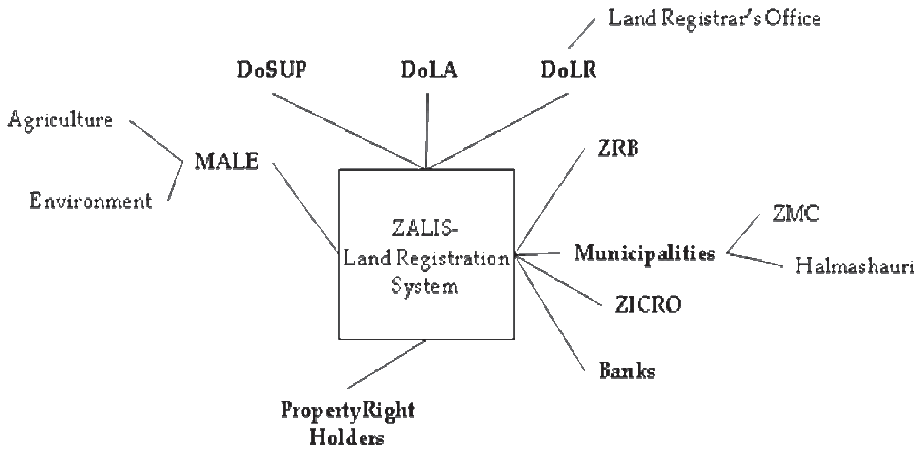


FIGURE 41 LRS and important stakeholders

Another diagram depicted in a similar way first inputs and outputs, and a third one listed the main hardware and software components. These diagrams were very simple and not worth to be presented here. Finally, the most immediate interfaces were shown in a separate diagram (FIGURE 42). The upper part lists high level interfaces like human actors and business processes. The middle part, peer level, shows other information systems, and bottom part is for lower level network and device interfaces.

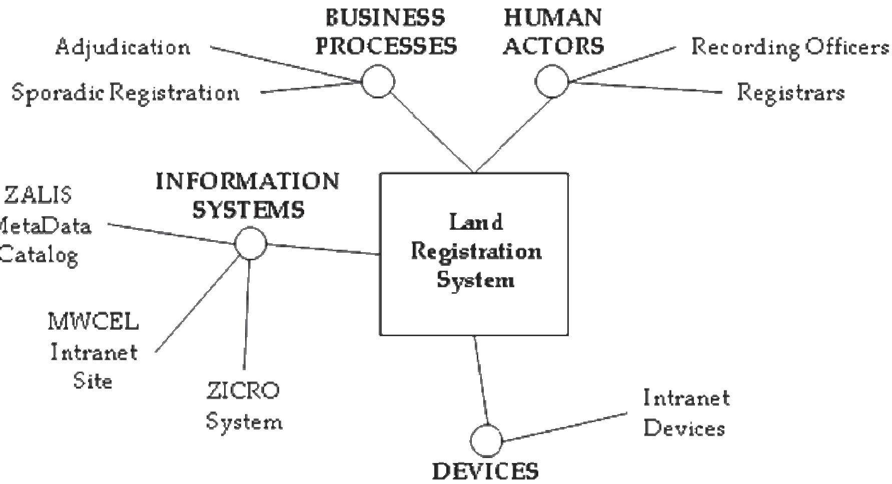


FIGURE 42 Immediate LRS system interfaces

The third case study *reinforced the usefulness of seeing the system more or less as a "black box" in the early stages of quality modelling*. This view underlines the understanding of quality as a relationship between the system and its environment (see the discussion in next chapter about the essence of quality). In addition, the *technical constituents and structures were again not of big interest to most of the informants*.

As was noted in connection with previous case studies, *the meta-model did not as such give guidelines on how to structure the description of environment*, what to include and what to exclude. In line with that, the initial views of context (FIGURE 41 and 42 above) in the third case study comprised only the most important actors, business processes and systems. Understanding of the relevant environment, however, grows gradually in the course of quality modelling.

Domain

General quality modelling was started with giving a questionnaire to the actors. It contained a diagram (FIGURE 43) depicting 24 different areas of concern, i.e. a predefined collection of domains. No quality attributes were listed in the diagram. This arrangement was based on the observation in previous case studies that *it is easier for the informants to think first only about domains and thereafter deal with individual attributes*. A glossary of domain names (APPENDIX 7) was attached to the questionnaire.

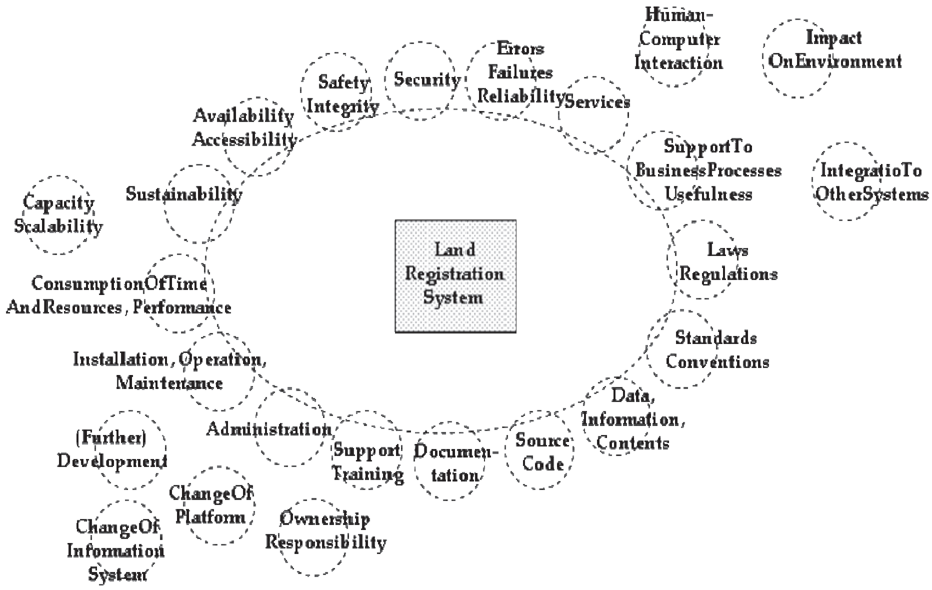


FIGURE 43 Collection of domains used in LRS questionnaire 1

The collection was created by the researcher and it was developed from the collections used in previous case studies. Related (according to researchers understanding) domains were placed near each other.

Priority of Domains

15 actors where asked first to prioritize at least ten domains. FIGURE 44 shows the result, the top seven domains. The height of column corresponds to the inverse average ranking of domain in question (the higher the column the higher the priority). 14 would have been the highest possible score (meaning that all informants have ranked the domain in question as number 1) and 0 the lowest. The entire results are shown in APPENDIX 12.

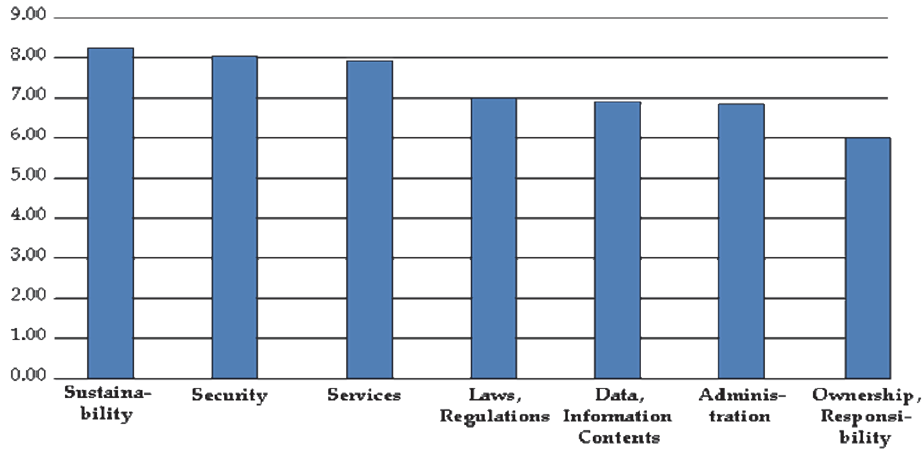


FIGURE 44 Top seven LRS domains

Like in previous case studies *it was difficult for some actors to rank the big number of domains*. Areas of concern seemed for them to be equally important. All selected actors were, however, persuaded to do the prioritization. Another comment was that domains belonged to different categories and were not comparable. This is true in the sense that some domain names refer to activities, like installation, some to relationships, like integration to other systems, and so on. Later on it was realized that the "services" and "support to business processes" are practically identical as areas of concern. Consequently, if they were presented together in the questionnaire the combined ranking would have been substantially higher. Based on this observation, these domains were handled together in the attribute model.

If the domain ranking of LRS case study is *compared to the domain rankings of previous cases, differences can be found*. In the LRS case sustainability and security were the two top domains, whereas in EMIS and website cases the top concerns were support to business processes and human-computer interaction. Having sustainability as number one complies well with the description of Africa as an environment for information systems given in section 5.1. Support to business processes came to third place in LRS case, while data and contents got this ranking in the other cases. Performance and integration to other systems were not among the top seven in LRS case, whereas laws and regulations were missing in the two other studies. These major *differences are partly due to the context and businesses in question, partly to the type and architecture of systems*. In Zanzibar, for example resources are scarce compared to Dar es Salaam and Maputo, which explains the importance of sustainability. And land registration as activity requires security and compliance to laws more than EMIS or a website. LRS is used by a limited number of professional officers, whereas EMIS and a website are a systems used by a big number of people, which explains the high ranking of HC interaction issues. In his article about appropriate ICT for developing countries van Reijswoud (2009, 4) underlines such concerns as 'having added value', 'technical support' and 'sustainable design'. The top domains registered in the three case studies confirm this view. 'Support to business processes' and 'services' mean in practice added value to users and domains like 'deployment', 'maintenance', 'sustainability' and 'administration' are comparable to 'technical support' and 'sustainable design'.

Quality Attribute and Attribute Set

Next the top seven domains were put into focus. In order to find out quality attributes for the system specific set together with some definition elements, use cases and scenarios were taken up even earlier than in the two previous case studies. Four selected actors (out of the initial 15) were given a second questionnaire. They were asked, in each prioritized area of concern (domain), to identify business use cases, system use cases or scenarios that would disclose if the land registry system has high or low quality with respect to the concern in question. For each use case or scenario the actors had to list positive or negative facts, experiences, situations, events, etc., that would indicate the existence or the lack of quality respectively. In addition, in connection with positive and negative facts they were asked to name a more specific quality attribute to which the fact can be related. To assist finding attribute names the questionnaire had as an attachment general attribute collections from McCall et al. (1977), Boehm et al. (1978), Dromey (1996), ISO 9126 (2001), and the attribute sets from two previous case studies. Finally, the actors where asked to prioritize the positive facts and related quality attributes inside each domain. The positive facts correspond to the statements and propositions (definition elements) of two first case studies.

It became clear that *it is quite easy for actors to list positive and negative facts, but difficult to identify use cases and scenarios*. The facts did not, however, always belong to the domain under which they were given and had to be rearranged by the researcher. *Naming and prioritizing quality attributes in the questionnaire was also difficult for the actors*. This might partly be due to the fact that the used general attribute collections did not give definitions for the listed attributes. It might also have been better to pre-position attributes under domains, like was done in previous case studies. Another cause is the inadequate knowledge of English language. All in all, the researcher played an important part in the modeling process after domains were prioritized. This suggests that *quality modeling process needs an experienced person as manager-editor*. The need of guidance became clear already in previous case studies. The above mentioned difficulties correspond also to what van Reijswoud (2009) notes about actors in developing countries. According to him majority of users and policy makers have still a very limited exposure to actual use of ICT. Understanding of capabilities is based on media and hear-say, and expectations are often unrealistic.

Analysis of the questionnaire drew many domains from outside the top seven into focus and gave ideas how to slightly rearrange old ones. Visibility came up as an important concern related to availability and accessibility. Services and support to business processes are as domains in practice the same and can be combined in the quality model. The word "environment" was a bit confusing to informants in this case study because of the entire development project was named "Sustainable management of land and environment"! So it was better to rename the domain for example "impact on context", and the meta-model element “environment” as “context”. Finally, domain “laws and regulations” and domain “standards and conventions” should both be divided into two subsections: 1) concerning ICT and 2) concerning business respectively. Indicators of system’s compliance to laws governing business activities are actually indicators of system’s support to business processes that should be carried out according to these laws. Again, all this *reinforces the requirement of keeping domain-attribute sets all the time open for modification*.

Attribution and Definition

While analyzing the second questionnaire the researcher started to build individual attribute models. This meant thinking in fact at the same time about all the remaining elements of quality model: attribution, use cases, contributors, the desired system-context relationship, definition, how to measure the existence and degree of the quality in question and connections to other quality attributes. In the model prototypes *all qualities were attributed to the land registration system as a whole*. The positive and negative facts were processed into positive propositions that describe the desired relationships and consequently constitute a major part of the definition of the attribute in question. The *positive facts, corresponding to statements and propositions in previous case studies, were now called “indicators”*. In addition, an overall definition was generalized from the individual propositions and merged to the general definition in attribute set. Following sample propositions are taken from the security attribute model:

- Overall definition: “Only authorized persons can access the LRS (hardware, software, data).”

- Indicators: “Only named data-entry officers are able to change named contents respectively.” “Companies servicing hardware cannot access information stored in that hardware.”

The above modeling procedure led to *merging two previous meta-model elements “definition” and “related to” into one named (desired) “system-context relationship” having indicator and model as sub-elements*. Indicators are facts that indicate the existence of relationship. Model is discussed below under heading “Relationship”.

Analysis of sustainability indicators showed that this domain is a “super-domain” that groups together a set of important domains. Accordingly, it seems to be more practical to keep it outside the actual domain collection together with “feasibility” concern and let actors model them independent of other domains. Analysis of the indicators in services (support to business processes) domain, in turn, disclosed that it is a really wide area of concern which had not got enough attention. The mission statement given to project team was vague and general. It told only to expand an existing system prototype to serve the land registration activity. No clear functional requirements were set. These emerged one by one during the implementation, test and operation. All facts about laws and regulations were placed under services domain, because, as was mentioned in previous section, they were actually statements about how the system can help users to carry out business processes according to the laws.

Use Cases and Scenarios

Use cases and scenarios were for the most part derived by the researcher from the positive and negative facts in filled questionnaires. As was noted above in connection with attribute set and attributes, *most actors were not themselves able to name correctly use cases or scenarios*. The list below shows by areas of concern what was identified. SCE stands for scenario, BUC for business use case and SUC for system use case.

Sustainability:

- SCE1: Donor money and foreign consultants are no more available for covering the costs of LRS and supporting MWCEL (Ministry of Water Construction Energy and Lands) in running, developing and maintaining LRS.

Security:

- SCE1: People working in or visiting the departments try to enter server room.
- SCE2: People outside or inside MWCEL intranet try to access LRS and its services.
- SCE3: LRS hardware or OS has to be serviced by private companies.
- SCE4: LRS data files are copied and taken outside the MWCEL intranet.

Services, Support to Business Processes:

- BUC1: Sales of property.
- BUC2: Land lease.
- BUC3: Building permit.
- BUC4: Planning water distribution network.
- BUC5: Planning and budgeting departmental activities.
- SCE1: Certain laws and regulations govern the activities of MWCEL and its departments.

Data, Information, Contents:

- SCE1: MWCL and different stakeholders start using information in LRS as basis for their activities.

Ownership, Responsibility:

- SCE1: Donor money and foreign consultants are no more available for covering the costs of LRS and supporting MWCEL in running, developing and maintaining LRS.

Administration:

- SUC1: System and database administrators use LRS and its administrative tools to perform updates, configurations, user and data management, etc.

Contributor

Contributor is a new meta-model element that was added to the meta-model, as described above, based on research findings of the first two case studies. In these case studies contributors were understood as system constituents and structures that directly or indirectly affect the quality relationships. The LRS case, however, pointed out that *contributors can also be found in the system context*. This is noticed for example by Voas & Agresti (2004) who mention as an example new security threats that can even quickly change the measured value of system security. Following are some of the contributors to sustainability found in LRS case:

- Internal: 1) development platform, 2) software architecture.
- External: 1) availability of ICT professionals in Zanzibar and 2) availability of financial means at the ministry for ICT (running, developing and maintaining LRS, and securing electricity).

As another example, in the data/contents domain some of the contributors found for qualities correctness and being up-to-date are:

- Internal: 1) constraint/validation mechanism (correctness) and 2) alert/reminder mechanism (being up-to-date).
- External: 1) system users understand the importance of data quality and are committed to take care of it (correctness) and 2) systems and practices for identifying and registering persons and organizations in Zanzibar (being up-to-date).

Relationship

Previous case studies suggested that characteristics which are commonly called quality attributes refer in the end to desired relationships between the information system and its user or other parts of the context. And this seems to be the essence of quality. Therefore in LRS case study, after formulating the positive definition statements (indicators) and finding out the contributors, *a general definition was given to the relationships described by the indicators*. It consisted of identifying the members of the relationship and relevant additional features and was *named "model"*. Sustainability, for example, got following description:

> "Sustainability is in first hand a relationship between LRS and the organization(s) accountable for it. In addition it points to the relationship between LRS and its users, developers, administrators, etc. A wider context, like national infrastructure (shared human, technical and information resources), must also be taken into account."

Security, in turn, was defined as follows:

> "Security is in first hand a relationship between LRS and different human or non-human actors in its context. It's about denying, allowing or delimiting visibility, access and connection between the parties in order to prevent something unwanted happening."

Abstract features of relationships can be depicted by ER- or domain-model type diagrams showing the members (entities), associations (names and directions) and some attributes. FIGURE 45 is a simple example taken from security domain. It shows three entities: the land registration system (LRS), one human actor and one external connected information system. One security related attribute (security mechanism) is attached to LRS and another (ICT skills) to the human actor. When the human actor or external system tries to connect to LRS the latter either denies or allows the connection and shows and hides information according to implemented security rules.

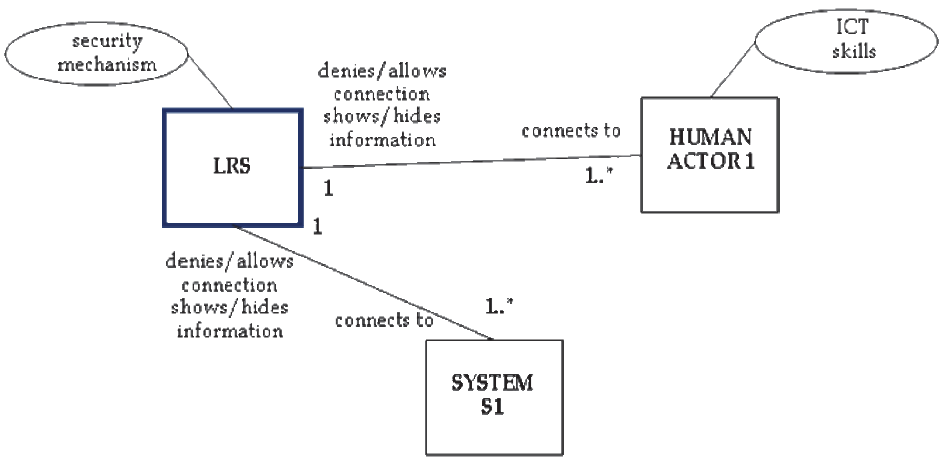


FIGURE 45 Abstract model of LRS security

Indicators, model and the verbal description of relationship can be used to refine the general definition of attribute in attribute set.

Connected to

Connections between attributes came into focus together with internal and external contributors. Like in EMIS case it was realized that *high quality in one respect presupposes high quality in some other respects too*. Sustainability as a "super-quality" was a typical example. If supporting attributes like the quality of data or usefulness are low, the sustainability is also weakened. Some of the created attribute models indicated connected domains instead of attributes. The latter means that if the quality with respect to certain area of concern is low it affects attributes within other domains. Poor documentations for example affects manageability, ease of administration, usefulness, sustainability etc. Only contributing connections were described in first attribute models.

FIGURE 46 shows relationships of top five domains together with priorities. The arrow above the matrix shows the direction of relationship and numbers equal to domain numbers in the pie chart. The angle of the slices represents the relative priority value (as a number in brackets) of attribute. The five original priority values were summed up and then each original value was divided by the sum to get the relative value. "+" sign stands for a positive contribution.

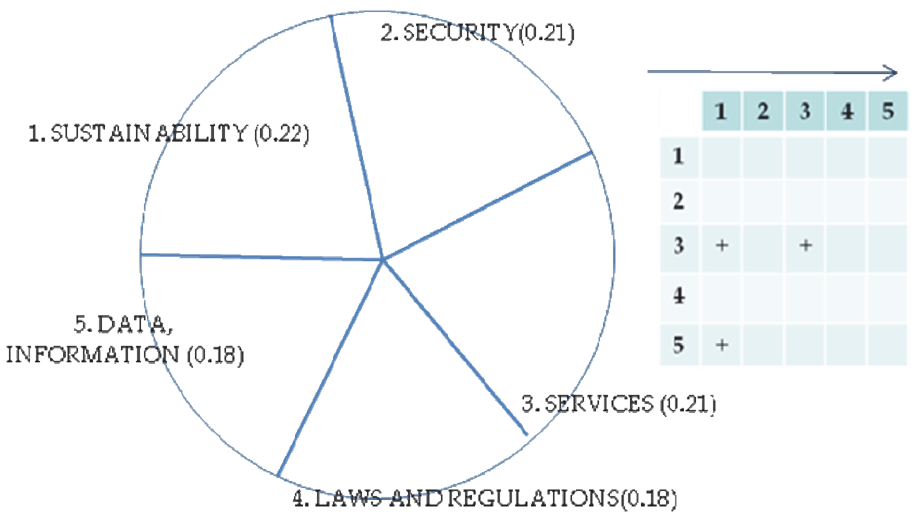


FIGURE 46 Priorities and positive contribution between top five LRS domains

Measurement, value and relativity

At the time of writing this thesis no actual measurements of LRS quality were made and the system version was 0.8. To each indicator in the attribute models was, however, attached a general suggestion for measurement method. For example it was planned that in definition section above mentioned security indicators “Only named data-entry officers are able to change named contents respectively” and “Companies servicing hardware cannot access information stored in that hardware” will be measured through a test. The testing can be outsourced to a company that is specialized in security issues. Some indicators like “LRS supports all business use cases and helps to achieve financial and other goals” (usefulness) can be measured by interviewing relevant actors, and some like “LRS prevents users to act against the laws and regulations governing MWCEL, and LRS functionality is error-free and complies with laws and regulations” partly by calculating the number of related constraints and alerts in the software and the number of errors that have been recorded per certain period of time. Relativity of quality definitions and measurements has been taken into account by indicating in attribute models use cases, scenarios and actors who designed individual indicators. Goal values were not set. FIGURE 47 exemplifies the diagrammatic representation of attribute models produced during the case study by depicting some of the main features of ‘usefulness’, ‘suitability’ and ‘being a watchdog’ attributes. The models were less graphical than in first two case studies. Relevant use cases and scenarios are listed on top of the diagram. Next indicates that 5 selected informants having mixed views of the system and environment were responsible for the modelling. Then follow the domain in question and the attributes in focus. On the left side of the diagram are sample indicators (actor numbers A1 etc. in brackets) and on the right side positively contributing domains and external contributors. Inside the rectangle is a list of anticipated internal contributors and below the rectangle a general definition of the relationship.

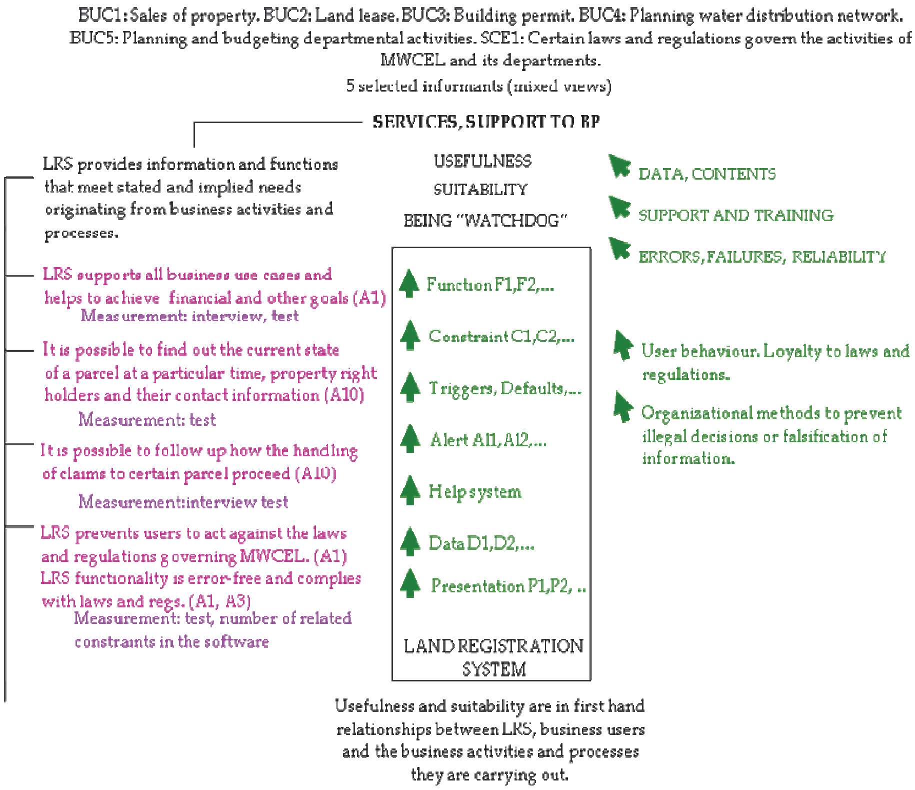


FIGURE 47 Simplified and combined graphical presentation of 'usefulness', 'suitability' and 'being a watchdog' attribute models

Being aware of the "pre-release" nature of the LRS system and that no actual quality measurements were made, the researcher presented for discussion, before closing the case, a rough estimate (FIGURE 48) of the degree of realization of quality goals in top areas of concern in November 2010. Height of columns in the diagram reflects following facts:

- ICT laws and regulations do not exist in Zanzibar.
- LRS is heavily supported by donor organization and sustainability, ownership and responsibility issues have not yet been solved.
- Implementation of contributors to security and ease of administration has been started.
- Support to business processes is still quite low because service requirements have not been fully defined.
- Quality of contents has started to get more attention because of public pressure to get correct land registration and land lease certificates from the system. Remaining problems in this domain are mainly due to external contributors like survey procedures and practices for identifying persons and organizations in Zanzibar.

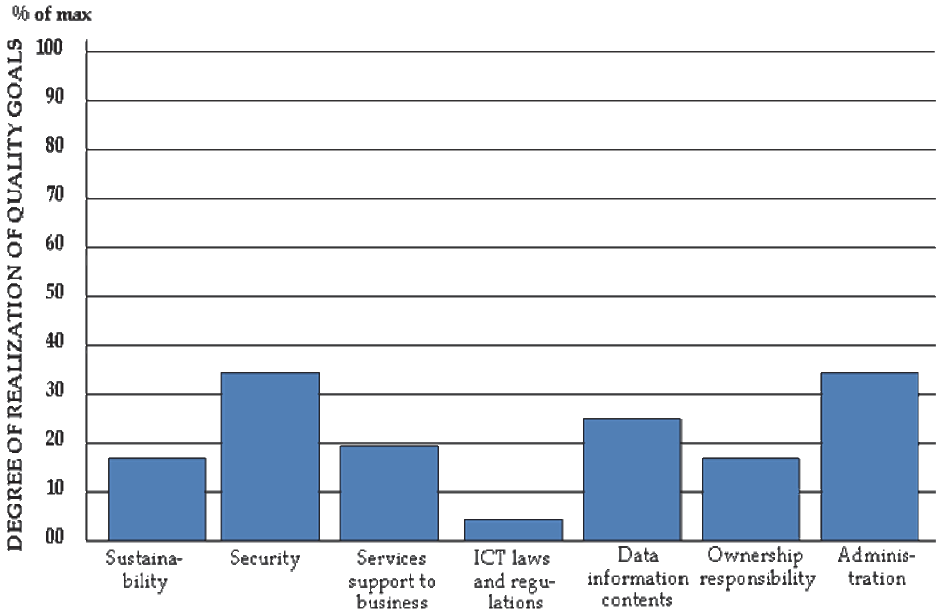
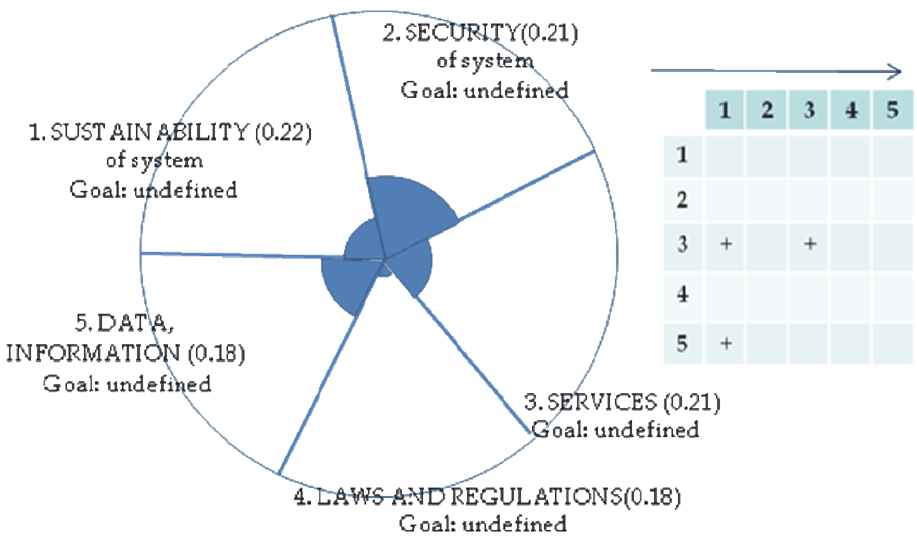


FIGURE 48 A rough estimate of the realization of LRS quality goals in top areas of concern in November 2010

FIGURE 49, in turn, gives a similar graphical view of overall quality design (with respect to top five domains) and its realization than was used in EMIS case. It shows priorities, attributions (the system as whole), goal values (undefined in this case), realization of quality requirements as percentage of maximum, and domain relationships.



	1	2	3	4	5
1					
2					
3	+		+		
4					
5	+				

FIGURE 49 Fulfilment of LRS quality requirements in November 2010

5.4.4 Concept of quality

The two first case studies drew attention to the essential aspects of information system quality, namely 1) desired relationships between system and context and 2) contributors. These studies also helped to differentiate between quality requirements and other requirements. The third case study confirms these findings. In addition it brings into focus actors' perspectives and views as determinants of how quality is defined and assessed. Accordingly the third distinctive feature of quality requirements can be slightly rewritten:

> - The priority, definition and measured level of implemented quality requirements can be relative to use case or scenario, *actor's perspective and view*, or some other features of the context.

5.5 Actor evaluation of models

APPENDIXES 13 and 14 show poster-like overall views of EMIS and LRS quality models respectively. Poster represents only one possibility to keep in actors' minds the main entities they have to think about in order to manage the system's quality. It shows the overall quality design, current status of quality, actors heard during quality modelling process (upper part), overall system architecture and important elements of context (left side). Individual attribute models cannot fit into the poster, but in the overall design part some acute issues can be notified with short text. It helps actors to reflect holistically on quality and continue the quality assurance process. At the end of EMIS and LRS case studies an overall evaluation of respective quality model prototypes was carried out. Assessment was made with respect to understandability, simplicity, coverage and usefulness of the models. Coverage was defined as the ability of the model to deal with all necessary aspects of quality. Evaluation was given on a scale of 0 to 9. In EMIS case three selected informants gave their opinion and in LRS case 15. In EMIS case the questionnaire was filled in interviews together with the researcher, in LRS case the questionnaire was filled anonymously. In the LRS case the evaluation was made in a workshop were the system itself and the quality model was presented to the participants. FIGURE 50 shows the average scores in both case studies. Filled columns represent EMIS case.

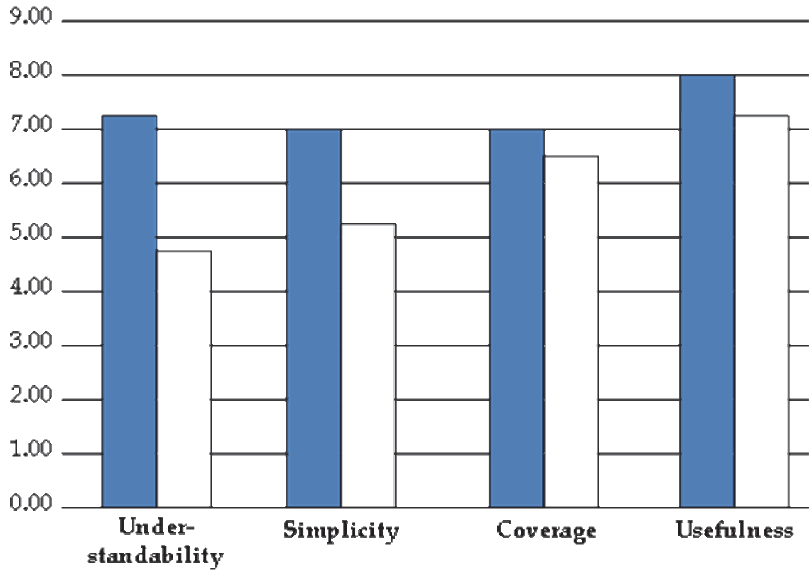


FIGURE 50 Results of overall quality model evaluation in EMIS and LRS case

The higher averages for understandability and simplicity in EMIS case may be explained by the fact that in the personal interview the researcher could better explain the models to actors. Further, the low figures for these two attributes in general may be explained by the fact that among the evaluators were no developers. Ordinary actors, especially in African context, obviously feel this kind of modeling difficult. All in all the figures prove that the quality models were experienced fairly useful. It would have been interesting to go deeper behind the individual evaluations. What aspects, for example, were according to evaluators missing from the models? The questionnaire was, however, designed to be general and in the LRS case even anonymous.

6 IMPLICATIONS OF CASE STUDIES

Previous chapter gave an account of the three quality modelling case studies in Tanzania and Mozambique. This chapter summarizes the implications. The discussion is arranged according to the detailed research questions posed in chapter 3. It ends with a section presenting the steps of quality modelling and relating them to general system development workflows.

6.1 Usefulness of artefacts and actions

Usefulness of artefacts and actions based on the meta-model give indirect validation to meta-model itself. The potential users of quality model can be divided into two categories: 1) the developers who will implement and test the quality requirements and 2) all other stakeholders who have participated in setting the requirements and afterwards can assess if these have been met. In the website and LRS case researcher represented the developer category.

Q1 Can the system specific quality models and the modelling process to be considered as useful and satisfactory?

Following observations indicate usefulness of the model, its elements or the modelling process:

- Awareness of numerous actor and stakeholder categories helped to realize how diverse the quality expectations can be and to recognize the relativity inherent in quality models and measurements. In fact every model feature can be traced back to particular actors and actor groups. (Cases 1, 2 and 3) Noting their perspectives and views, in turn, made the expectations understandable. (Case 3) Further, paying attention to actor element resulted in getting different actors involved in the quality modelling process and development process in general. (Cases 1, 2 and 3)

- Depicting the information system first as a "black box" surrounded by elements in the context helped actors to focus from the start on the desired system-context relationships as the essence of quality. This effect can be fortified by trying to give each structural-behavioural element in the more detailed views of the system a raison d'être through the same desired relationships. These practices are, however, more a feature of meta-model application process that the meta-model itself. (Cases 1, 2 and 3)
- Paying more attention to system context than traditional quality models do is essential for understanding system quality. (Cases 1, 2 and 3)
- Grouping quality attributes by domain (area of concern) helped to understand individual attributes. One and same attribute name can even occur in multiple domains with different meanings. (Cases 1, 2 and 3)
- Predefined general attribute collections helped to identify and select quality attributes for system specific models and to find out biases or gaps in quality design. The attribute collections as such are not, however, part of the meta-model. (Cases 1 and 2)
- Prioritization helped to focus on qualities that are most important to the business in question. Spreading limited resources over a large number of concerns with different significance could result in a low overall quality. On the other hand, focusing even on a fairly small number of key attributes can have a substantial effect on product quality. Actors were also very interested in the results of prioritization. (Cases 1, 2 and 3)
- Use cases and scenarios are indispensable in defining quality attributes. They helped actors to create statements (indicators) expressing the existence of a particular quality. It was also useful to differentiate between business use cases, system use cases and scenarios. Linkage to use cases disclosed also that one and same system level quality can mean different things in connection with different use cases or be valid only in some contexts. (Cases 1, 2 and 3)
- Categorized definition sentences, called relationship indicators in the final meta-model, constituted a valuable part of attribute models. They made qualities "tangible" and revealed the nature of qualities as desired relationships between system and its context. In addition they formed a basis for metrics. These sentences were written in ordinary language and were understandable to all stakeholders. Indicators can be appended with actor names and use case identifiers in order to remind of the relativity inherent in the model. (Cases 1, 2 and 3)
- Identifying on general level internal and external contributors was crucial to realizing the quality expectations. Anticipating them early in the development process helped designers and programmers. Indicators and contributors belong to the core elements in each attribute model. (Cases 1, 2 and 3)
- A general or abstract model (part of the relationship element in the final meta-model) of the relationship helped to figure out the type and nature of each relationship. (Case 3)

- All in all, the individual attribute models guided actors to define qualities correctly and accurately. (Cases 1, 2 and 3)
- Describing attribute connections, especially supporting and conflicting relationships, helped to finalize the overall quality design. (Cases 1, 2 and 3)
- Metrics was not in the very focus of this study. However, the simple measuring arrangements that were devised and measurements that were carried out proved to be useful and helped to refine the quality model. Considering metrics lead also to think about the relativity of attribute definitions, prioritizations and measured values. (Cases 1, 2 and 3)

Observed inadequacies and difficulties, in turn, were following:

- In the initial meta-model actor element was positioned inside attribute section and it included non-human actors. These issues were rectified in the intermediate meta-model. (Cases 2 and 3)
- It was difficult in practice to reach and contact all relevant actor categories. Missing perspectives may, therefore, weaken the instantiated quality models. (Cases 1, 2 and 3) The initial and intermediate meta-models did not include these perspective and view as elements. In the third case study actors' perspectives and views were listed but not orderly described. (Case 3)
- There was no explicit theory or model of environment behind the context descriptions in quality models. (Cases 1, 2 and 3)
- The number of attributes and domains in attribute collections was felt by some actors too large. Understanding the meaning of domain and attribute names was also often difficult. (Cases 1, 2 and 3)
- It may be difficult to find a perfect, widely accepted and logical set of domains for grouping quality attributes. (Cases 1, 2 and 3)
-In initial meta-model priority element was positioned inside attribute section. This issue was rectified in the intermediate meta-model. (Cases 2 and 3)
- Prioritizing large domain and attribute sets was felt difficult among actors. In some cases all domains and attributes were felt to be equally important. Organizing a workshop can help in this respect. (Cases 1, 2, and 3)
- Qualities in the resulting models were mainly attributed to the system as whole. From lower level constituents only user interface was noticed. (Cases 1, 2, and 3)
- Sets of definition sentences and indicators were incomplete in all the instantiated models. (Cases 1, 2, and 3)
- Deriving use cases and scenarios, and relating indicator facts to attribute names was difficult for most actors. (Case 3)
- Orderly use case descriptions were for the most part missing. (Cases 1, 2 and 3)

- Contributors were not sufficiently described in the prototype quality models. (Cases 1, 2 and 3)
- Attribute connections were not fully described in cases studies and goal values are missing from quality models. (Cases 1, 2 and 3)
- Measurement of attributes values was not exact and detailed. (Cases 1, 2 and 3)

Q2 Does the use of meta-model cause bearable amount of overhead in terms of time and other resources?

Comparing the quality modelling process to overall development process, starting with requirements capture and continuing through analysis and design to implementation, shows a close similarity. There are actually no extra phases to go through or steps to be taken. What makes difference is the approach and focus. In quality driven development the process is led by prioritized and well defined requirements concerning the relationships between system and its context. And the quality model itself is just the core part of requirements model. These aspects are discussed below in section 6.6.

Conclusion

The system specific quality models created in the course of three case studies, and the process itself, were evidently useful to actors in many respects. The inadequacies relate for the most part to practical difficulties in communicating with actors, weaknesses in the used domain collections, and leaving some parts of the models unfinished due to limited timeframe. Defects found in the actual meta-model are fixed in the final version. Accordingly, despite of the numerous minor difficulties listed above, the positive observations prove the general usefulness of created system specific models and consequently validate for their part the proposed meta-model. See also overall model evaluation made by actors in section 5.5 above.

6.2 Conformity with real world

Like usefulness of artefacts and actions, conformity of system specific models to real world gives indirect validation to meta-model itself.

Q3 How well do system specific model elements represent the real world items and diverse data in the case study contexts?

Following observations indicate conformity of the models to real world items in the context:

- Having actors, their perspectives and views as model elements connects the whole model more firmly to real world. (Case 3)
- Viewing reality as things (objects) having features (attributes) and categorizing (domains) features are normal cognitive processes. Attribute set, in turn, is a real world item as an artefact of group work created by the development team. (Cases 1, 2 and 3)
- Attribution (a certain quality required from certain system constituent rather than from the system as a whole) means just a more detailed adherence to reality. (Cases 1, 2 and 3)
- Prioritization, ordering activities and setting goals are a common and real traits of human behaviour. It is usually caused by lack of resources or conflicting goals. (Cases 1, 2 and 3)
- Breaking down the concept of use case into three components (system use case, business use case and scenario) increased adherence to reality.
- Use cases, scenarios, definition statements and indicators refer to very concrete real world phenomena. (Cases 1, 2 and 3)
- Contributors are real elements of information system or its context. (Cases 1, 2 and 3)
- The "two-layered" definition of quality, in first place as a desired system-context relationship and in second place as internal and external contributors, conforms more to the real world than those alternatives that tend to equal quality to internal characteristics of product. (Cases 1, 2 and 3)
- Attribute connections reflect the way how connected attributes' external and internal contributors as real world elements are related to each other. (Cases 1, 2 and 3)
- Abstract models of system-context relationships are generalizations from real world phenomena. (Case 3)
- Metrics conforms to reality if it measures correctly real world phenomena, namely contributors and facts indicating the relationships. (Cases 1, 2 and 3)

A few potential nonconformities, in turn, were:

- Every information system that is in use is tied to its context. Initial separation of context and system, and reducing the latter to a set of almost meaningless constituents and structures, was used in quality modelling, but only for heuristic purposes. Paradoxically, by trying to separate things and deprive their meaning one can discover both. (Cases 1, 2 and 3)
- Every model element that needs extensive sub-modelling, like context, is suspect for not conforming entirely to reality with respect to its components. (Cases 1, 2 and 3)

- Domain means a particular field of thought, activity or interest. The big challenge is to find a commonly acknowledged set of domains conforming to reality and relevant to information system quality. On the other hand, a domain and domain set, like attribute set, can be taken as a real artefact of group work. (Cases 1, 2 and 3)

Conclusion

In general the system specific quality models created in the course of three case studies conform to real world, which supports the validity of the meta-model in that respect. The conformity was increased by refining some model elements like actor and use case. The reality of large model elements, like context and domain set, depends on the reality of their components.

6.3 Comprehensibility

Comprehensibility of system specific models reflects the comprehensibility of meta-model and validates the latter in that respect. Like usefulness the comprehensibility of models must be assessed by their principal users, i.e. on one hand developers and other relevant stakeholders.

Q4 Are the system specific models comprehensible and do they make sense to actor-informants?

Many of the positive observations listed in section 6.1 above indicate comprehensibility of models as well. The following are some additional points:

- At least the black-box or overall view of information system that was used in the case studies was not too demanding for ordinary actors. (Cases 1, 2 and 3)
- For some actors it was a positive discovery to understand information system quality in first place as a desired relationship between system and context instead of just assessing what is inside the system. (Case 3)
- Context elements taken into models were proposed by the actors themselves which guarantees comprehensibility. (Cases 1, 2 and 3)
- Linkage to use cases and scenarios made qualities more tangible and comprehensible to actors. (Cases 1, 2 and 3)

Some observed difficulties listed in section 6.1 above indicate incomprehensibility of models as well. Following are again additional points:

- Actors felt that large domain and attribute collections are difficult to read and understand. This can be partly solved by first separating the domain

collection from attributes. Splitting the domain and attribute collection and attribute selection task accordingly into smaller parts (by sub-system, use case, etc.) is also helpful. (Mainly cases 1 and 2)
-Some actors felt that domains in the collection belonged to different categories and were not comparable. (Case 3)
- Ordinary actors were quite unfamiliar with information system quality attributes and needed much guidance. Consequently ICT professionals in the development team together with the author of quality model had in practice a bigger influence in this respect. (Cases 1, 2, and 3)
- The relationship between domain, use case, scenario and individual attribute was not clear to some actors. (Case 3)

Conclusion

In general the system specific quality models created in the course of case studies were comprehensive to actors. Some concepts cause, however, difficulties to ordinary actors. This may indicate that the meta-model itself is comprehensible enough only to developers and other actors experienced in system modelling. The difficulties experienced by actors are reflected also in the fairly low scores given to understandability and simplicity in overall assessment of models (see section 5.5. above).

6.4 Comprehensiveness, generality and flexibility

Comprehensiveness, generality and flexibility are meta-model attributes that can be evaluated by a system developer who is leading the quality modelling process and creating the model. In this study the researcher had to take this role.

Q5 Is the meta-model comprehensive but at the same time distinctive enough?

Q5a Does the meta-model cover all essential aspects of information system quality?

The initial quality meta-model was as such more comprehensive than any of the models found in the theoretical background. In the course of case studies it was completed with three additional elements (contributor, perspective and view) and a number of sub-elements. This does not, of course, prevent further research from disclosing gaps in the meta-model. The coverage evaluation given by actors (see section 5.5 above) is not very reliable.

Q5b Does the meta-model guide in modelling overall quality as well as individual quality attributes?

Overall quality can be clearly defined on basis of the meta-model. It is embodied in the categorized and prioritized attribute set describing the interrelationships of attributes and setting goal values for them. Attribute sub-model, in turn, guides modelling individual attributes.

Q5c Can quality attributes be differentiated from requirements in general using the conceptual framework?

The neutral term "requirements" has in literature quite broad meaning. Jacobson et al. (1999, 448), for example, define requirement as "a condition or capability to which a system must confirm". Maciaszek & Liong (2005, 16) define user requirements as "statements of what services the system is expected to provide and the constraints under which it must operate". Ingram (2009, 161) states first that "requirements are a way of specifying what the application needs to do". Later Ingram (2009, 163) writes that "requirements map directly to the quality characteristics" and gives as examples of requirements things like "reliability", "performance", "availability" etc. Based on the analysis of the two case studies there is no exact or "black and white" *way of differentiating between quality requirements and other requirements* or requirements in general. *A set of distinctive features can,* however*, be suggested*:

- Quality requirements refer in first place to expectations about the relationships between information system and its context. In the second place quality requirements refer to those immediate things, like contributors, that are needed to realize the expected relationships.
- Quality requirements are very important to actors and have priority over the rest of requirements. Further, actors set goals regarding to what degree the requirements need to be met.
- Priority, definition and measured level of implemented quality requirements can be relative to use case or scenario, actor, or some other feature of the context.

Q6 Is the meta-model general enough to be applicable to a variety of contexts?

The meta-model has been successfully applied to three different information systems in three different contexts. This, however, is still only a weak indication of wider generality.

Q7 Does the use of meta-model provide flexibility in quality modelling without loosing essential aspects of quality out of sight?

According to answer to question Q5a above the meta-model covers essential aspects of quality. At the same time it does not prescribe any selection of actors, domains, attributes or priorities. Prepared domain and attribute collections are used only as reminders or check-lists. All stakeholder perspectives and views are in principle of equal value. In addition it does not prescribe any particular methods to be used in instantiating the meta-model elements. (Cases 1, 2 and 3)

Conclusion

The three case studies increased the comprehensiveness of the meta-model with respect to modelling both overall quality and individual attributes. It was also found to be general and flexible enough for these three contexts. Three cases are, however, still a quite weak indication of comprehensiveness, generality and flexibility.

6.5 Comparison to background theories

Q8 Does the meta-model describe and explain the information system quality and quality models more comprehensively and sensitively than background theories? (Järvinen 2001, 32)

A comparison of initial meta-model to theoretical background was made above in section 4.3 and the limitations of the latter were discussed in section 3.1. The comparison proved the meta-model to be more comprehensive and address in general well the critics listed in section 3.1. Only two issues were left open: 1) guidelines to carry out a measurement of overall quality and 2) the difference between quality requirements and requirements in general. Both issues have got a solution through analysis of the case study finding. Refined definitions for information system quality and quality requirements as well as for overall quality will be given in chapter 7 as part of the final meta-model. What comes to detailed metrics and formulas for calculating overall quality, these are not subject of the study.

6.6 Using the meta-model in connection with development process

Q9 What are the implications of case studies for using the meta-model in connection with the development process?

One of the arguments for embedding a case study as part of the EMIS development project in Tanzania was that information system quality must be dealt with as an issue of its own and as a separate component. Otherwise quality considerations would be easily forgotten in the midst of pure technical challenges. A similar view is taken, for example, by Boehm et al. (1978) and the process or component was called "quality assurance activity". Nielsen (1993), in turn, who dealt with one aspect of quality called "usability", wanted to see usability engineering as an activity that takes place throughout the product lifecycle starting as early as during market research. He pointed out the financial impacts of quality (usability) and underlines that substantial usability engineering resources must be allocated. (Nielsen 1993, 71-73, 81-85) The three case studies suggest, however, that even this kind of arrangements do not guarantee that time is properly taken for quality issues. Tight time-frames, geographical separation of quality workers and designer-programmers, weak project organization, lack of motivation on part of software provider, and many other factors can damper the results. Even the slightest overhead and ancillary work is often resisted. The effect of these process related factors is noted e.g. by Boehm et al. (1978) and Tian (2004). Boehm et al. (1978) propose as a remedy the use of automation in quality assurance to get final payoff in terms of reduced verification testing and fewer errors. But that is again a time consuming arrangement and only a part of the solution.

During the last ten years the software developer community has seen a rise of different approaches characterized by the word "driven". The object-oriented camp has called their method "use case driven" (Jacobson et al. 1999). Crispin (2006) advocates "test-driven" development as indispensable. Huang & Boehm (2006) are in favour of "economics-driven" software or value based software engineering. Quality in different forms appears also as a driving force in writings. Chung et al. (1999) put up non-functional requirements as the motor for software design. De Bruin & van Vliet (2003) call their approach "quality-driven approach to software architecting". Similarly Niemelä & Immonen (2007) call their work "quality-driven architecture design". Denger & Shull (2007) explain in their article what "quality driven inspections" mean in connection with system development. The experiences in all three case studies presented in this thesis suggest introducing still another approach that could be called simply "quality driven development". It does not only mean that quality design and implementation must be a truly integral part of software development that cannot be given up. It means raising the quality modelling from the role of being just a separate component to the status of an umbrella like driving force. If stakeholders in the end want a quality system, quality is also the issue to start with. All other goals and decisions should be subordinate to it and all other design elements in line with quality design. In this sense quality drives the whole development process, not only, for example, architecting. And if quality is understood as a set of desired and most essential characteristics of an information system defining its relationships to environment, the question of overhead and extra work for developers looses its meaning. Qualities equal the core requirements and not something peripheral like they seem to be, for example, in traditional object-oriented methodology. Quality driven development challenges all the "rapid and dirty" approaches where system development starts with gathering quickly just functional requirements, i.e. what information is intended to put into the system, stored in it, how the information is processed and what are the different outputs, and where developers jump right away into design and implementation process.

FIGURE 51 shows the steps of quality modelling and how they relate to main phases of system development. The quality modelling process is here put, like in previous workflow figures (FIGURE 15 and 38), against the background of object-oriented software development method as presented by Jacobson et al. (1999) to exemplify how the overall development and quality modelling processes fit together. Requirements, analysis, design, implementation and test are the core development workflows in Jacobson's framework. Operation is added to underline that measuring quality does not end with the tests. The figure is almost the same as FIGURE 38 depicting the modelling process in third case study and steps again equal with meta-model elements. Element names "environment" and "attribute connections" have been changed to "context" and "attribute relationships" respectively, as well as the order of steps 4, 5, 7 and 8.

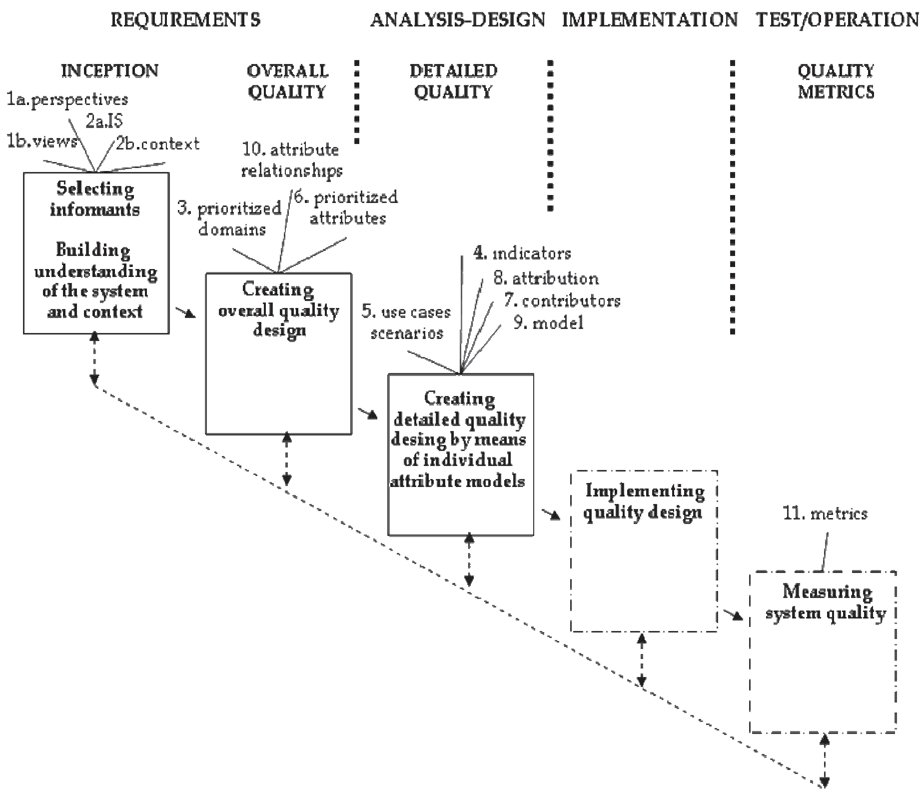


FIGURE 51 Activities and steps of quality modelling as part of system development

The first task during a phase that can be called "*inception*" is always formation of an actor group that carries out quality modelling. This group is so important that it is positioned as the first element in quality meta-model followed by the initial understanding of the system and its context. The latter things have an impact on the selection of first group members. Even the first understanding of the system is inevitably an interpretation made by some human actor(s). If the group is formed when the project is initiated and most of the members usually participate also into other development activities of the same system, it guarantees that quality modelling will be integral and dominant part of the whole software development process. As will be stated later in the chapter about final meta-model, the actor-informant group is properly formed if all essential actor-stakeholder categories are represented. Case studies showed, however, that it is difficult to engage all relevant stakeholders. Therefore, *it is practical to start the work with most immediate system users and expand the informant group later according to needs and possibilities*. If relevant, cultural aspects have to be taken into account when actor group is formed. Some studies in developing countries (e.g. Thanasankit & Corbitt 2000, using a Thai case) show that social structures and hierarchies can be tall. All kinds of decisions must be approved by managers or committees. This is true also to some extent in East-Africa. In these kinds of environments actor group must cover not only people with knowledge but also people with power. Before starting the actual quality modelling the informant group must be aware of the perspectives (step 1a) its members possess and the views (step 1b) they can have of the target system and its environment. After that comes the task of gaining initial understanding of system and context.

In all three case studies the system was first described (step 2) to informants more or less as a "black-box" or by simple structural diagrams. Only technical reports meant for developers and ICT personnel went further into details. The purpose of this view is to *turn attention from the very beginning to the desired relationships between system and its context*, i.e. quality requirements. According to the conceptual analysis and findings in case studies all structural-behavioural elements inside the system must in the end have their "raison d'être" in quality requirements. This resembles the Taylorian view where, for example, messages stored in information system have no inherent value, and the value of entire system emerges only within a context (see Scholl et al. 2011, 790). The description of information system will gradually become more detailed and transparent during "attribution" and "contributors" steps described below.

After initial system description follows the initial description of the system context (step 2). It is a more important task in the initial phase than description of the system itself. Entities in the environment, including different human actors, determine what is required of the relationships between system and context. Descriptions of environment created in the case studies were quite simple but served the purpose of finding the most basic relationships. *A suitable "meta-model" of context would guide in focusing on the most relevant features of environment* with respect to quality modelling.

After inception phase the *overall quality design* can start. It consists of creating a prioritized system specific set of domains (areas of concern) and quality attributes out of known alternatives. The results of inception phase together with the initial overall quality design form a kind of sketch of the target system in relation to context. In the first case study actors were given a predefined selection of domains and attributes and then asked to prioritize them. This was found to be difficult for some informants because of too many items to deal with. Therefore, in the second and third case studies domain collection was separated from the "palette" and actors were asked *first to prioritize the domains* (step 3) first and then attributes. At this stage conflicts may arise and a method for solving them must be found. This is discussed in a recent article by Holmström & Sawyer (2011, 36-37).

After agreement about priorities in two first case studies followed the process of finding out definition elements for attributes in connection with use cases and scenarios. This introduced *the detailed quality design*. In the third case study still a bit different order of steps was tested. After prioritizing domains the actors were asked to identify inside domains use cases and scenarios that would disclose if the system has high or low quality. Next actors were instructed for each use case to list positive and negative facts that would indicate the existence or the lack of quality respectively, and finally to name more specific quality attributes to which the listed facts could be related and prioritize them. Again the procedure caused difficulties to actors. What was most easily achieved was a collection of positive or negative facts. Use cases and attribute names were lacking almost totally from the filled questionnaires.

The above observations suggest that *after letting actors prioritize domains the most useful and productive step is eliciting positive and negative facts* (step 4) *indicating existence or the lack of quality inside each prioritized domain*. In the final meta-model these facts are called "indicators" and they constitute an important part of individual quality attribute models. The identification of use cases, scenarios (step 5) and individual quality attributes (step 6) related to the facts can follow thereafter, as well as prioritizing the attributes. Next comes looking for contributors (step 7) and possibly attributing (step 8) the qualities more precisely to certain system constituents. In these steps *actors need a lot of support from someone experienced in quality modeling*. The assumption that stakeholders are able to understand and communicate present and future needs in a clear way has been recently criticized for example by Holmström & Sawyer (2011, 35). Defining internal contributors is part of system model and consequently requires that developers take part in the process. The steps from 5 to 8 are in practice carried out rather simultaneously than one after another. Unless system component or sub-system specific attribute collections and sets are used, quality characteristics are usually at the beginning of quality modelling attributed to the information system as a whole. More specific attributions grow up during the modelling process, especially in connection with contributor element. All the steps of detailed quality design can potentially affect and cause changes in the overall quality design, i.e. the prioritized domain-attribute set.

After listing indicators and identifying use cases, scenarios and contributors there exists enough material for creating a formal representation of the relationship between system and context called "model" (step 9). At this stage indicators, model and its verbal description can be used to refine the general definition of the attribute in question in the domain-attribute set. The remaining steps are identification of attribute relationships (step 10), especially conflicts, and designing a procedure for measuring (step 11) actual attribute values in connection with testing and operating the target system.

The principle of flexibility and freedom was underlined in all three case studies regarding selection of system specific domain-attribute sets. There are, however, some core requirements that are commonly acknowledged to be important per se and should always be included in attribute sets. In the initial meta-model they were called “vital quality attributes”. First of all any information system must be feasible (before even trying to create or acquire it), available, accessible and sustainable. In addition, it must be useful and therefore frequently used or, in some cases (e.g. computer games), have an ability to entertain. All the other characteristics, usability in the front line, follow from or affect the before mentioned. They hamper or make it easier to use the information system, make it less accessible etc. FIGURE 52 depicts this view.

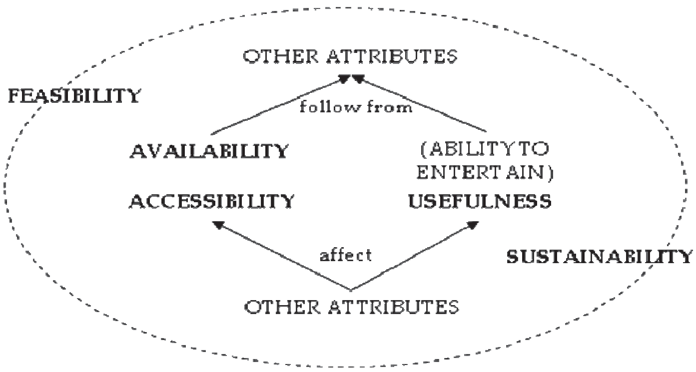


FIGURE 52 Core quality requirements

The above principle was not forced on actors in the case studies. In two first cases only usefulness (in other words support to business processes) was selected by actors into top five qualities, and in the third study both usefulness and sustainability. Analysis of sustainability showed that it is a kind of “super-attribute” or domain that actually groups together a set of attributes. Missing of the three remaining core requirements can be explained. Feasibility was not listed at all in the prepared domain and attribute collections. Accessibility and availability, in turn, might be felt kind of self-evident. Actors did not argue logically that if the system is not available and accessible it is in practice non-existent.

The division into core qualities and other qualities resembles the division into key quality factors and locally defined factors by Khaddaj & Horgan (2005) in their Adaptable Quality Model (AQM). The key factors are required of all products. Locally defined factors, in turn, apply only to the current product being developed. The difference is that AQM defines in total seven key qualities: maintainability, usability, cost/benefit, security, reliability, timeliness and correctness. And in fact only two of them, usability and correctness, coincide for the most part with usefulness in core requirements. The latter can be seen to represent even more basic requirements than key factors in AQM. In a recent article Buschmann (2011), in turn, underlines just usefulness (in his terminology "business suitability") and usability as the key requirements for software. He also recommends using scenarios that treat the system as black box in requirements gathering.

From FIGURE 51 one can see that overall quality design coincides with requirements capture in unified software development model. Jacobson et al. (1999) divide requirements into two categories: functional and non-functional requirements. Functional specifications tell what the system is supposed to do for the users. Non-functional requirements, in turn, correspond to what are traditionally called quality attributes, like performance, availability, etc. This means that the core quality 'usefulness' in fact covers the functional requirements in Jacobson's model. And consequently a prioritized system specific attribute set with goal values and attribute relationships, can cover the whole range of requirements and act as a guide and driving force for the whole development process. The importance of integration between functional and other requirements was seen for example by Kotonya & Sommerville (1996).

Conclusion

Case studies pointed out what is the most productive order to apply quality meta-model elements. They also helped break down the quality modelling activity into steps and embed them into the overall system development process. Qualities appear to equal the most desired system characteristics and concern the relationship between system and its context. Worries about extra work translate to worries about the most important work, namely modelling core requirements. Quality model can be the arch-model of system development in an approach labelled "quality driven".

7 FINAL META-MODEL

The initial version of meta-model was presented in Chapter 4. It was defined that a the meta-model is a kind of template for individual information system product quality models that “tells which things must be specified or modelled with respect to overall quality and individual qualities in order to be aware of them, to understand them and to be able to implement them”. Findings in the three case studies showed a need to make changes in the initial and intermediate versions of the meta-model. They also raised a need of sub-modelling, for example in terms of actors and system context. Following a summary of the implications of the three case studies in previous the chapter, the final version of the information system product quality meta-model is presented in this chapter. After an overall view, each model element is discussed separately. Questions for future research are occasionally appended. The fourth section deals with the concept of quality and quality requirements and the nature of system specific quality models. The final section evaluates the achieved conceptual framework in the light of alternative quality models presented recently.

7.1 Overall view

A comprehensive meta-model, that is needed to fully account for the quality of an information system and act as a template for lower level quality models, is a hybrid model (FIGURE 53) with six sub-models. 1) Human actors with their perspectives and views (symbolized by the filled triangle without borders) are actually part of an activity or a process (quality modelling or system development) model. Actors are typically seen as elements of activity models. To gain an understanding of the target information system on necessary levels of abstraction 2) an information system model is needed. C1 and C2 symbolize system constituents. To deal correctly with 3) the context one needs to model it to some extent. E1 symbolizes an entity in the context. In figure UC in “Business

UC" stands for a use case. Next, 4) the domain and attribute set reflects the core requirements and areas of concern for stakeholders, and can be viewed as a model of overall system quality. QA1, QA2, QA3 and QA4 in the figure are representing quality attributes. Roman numerals indicate the priorities of domains and the priorities of attributes inside the domains. Goal in the figure means a goal value set for a quality attribute. Individual 5) quality attribute models are in essence models of desired system-context relationships or states of affairs. Finally 6) the metrics part is a model by itself. It describes procedures and means of determining if the intermediate or the end product of a development process complies with the design in respect of quality. Developing the sub-models further is not in the focus of this study. Similarly, the individual techniques and methods (e.g. AHP that was used in the first case studies for prioritizing attributes) for implementing model elements as well as best practices of presenting (e.g. Kiviat diagrams used by Georgiadou (2003, 316)) the models and elements graphically or textually are outside the scope of this study.

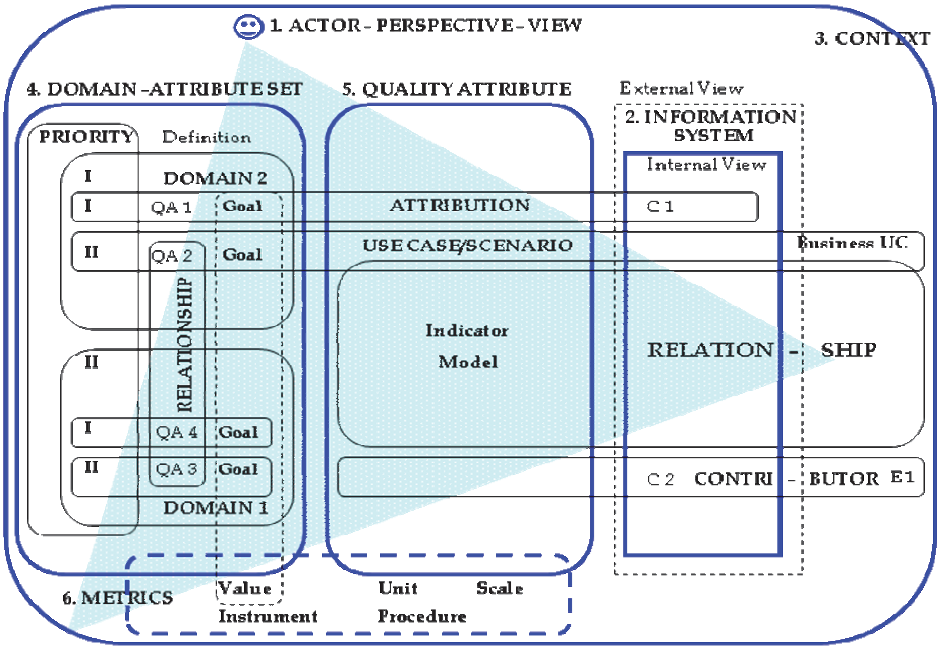


FIGURE 53 Diagrammatic representation of final quality meta-model

Domain-attribute set and attribute models are the core sub-models. The former is the model of overall system quality. In a strict sense, metrics is not an actual part of the product quality model. This is symbolized with a dotted rectangle line in the figure. It is a blueprint for measurement procedure, not for the product. It designs a procedure for checking if the end product of system development complies with its blueprint in respect of quality. The numbering in FIGURE 53 indicates a relative order in which sub-models come into focus during the quality modelling process. The overlapping of rectangles symbolizes:

- PRIORITY: Both domains and attributes are prioritized.
- ATTRIBUTION: A quality attribute (as QA1 in the figure) in an attribute set can be attributed to the information system as a whole or to some of its constituents (C1 in the figure). The attribution of qualities starts when the domain-attribute set is created and is reviewed during modelling of individual attributes.
- USE CASE/SCENARIO: Use cases comprise system and business use cases, the latter being part of the context. An attribute as (QA2 in the figure) can be initially connected to use cases already during the creation of a domain-attribute set. Attributes can have different order of priority in connection with different use cases.
- RELATIONSHIP (on the left hand side in figure): Quality attributes are interrelated in many ways.
- RELATIONSHIP (on the right hand side in the figure): The majority of quality attributes refer in the first place to desired relationships between an information system and its context. These relationships become visible and are defined in connection with use cases and scenarios.
- CONTRIBUTOR: Internal (as C2 in the figure) and external (as E1 in the figure) contributors determine for their part to what extent the desired relationships are met. Contributors are system constituents or things in the system context.
- VALUE/GOALS: Goals are target attribute values.
- METRICS: Metrics can be designed for individual quality attributes or overall quality (taking into account the whole attribute set).

The notion of 'hybrid model' above is in a way similar to 'multiperspective approach' or 'holistic quality models' discussed for example by Dahlberg and Järvinen (1997). The message in both cases is that information system quality cannot be reduced to a set of system characteristics and defined or assessed from only one or two points of view. Further, each member of the hybrid model is understandable only in the context of other members. A technical model of information system, for example, is in the end not understandable without a connection to actors, requirements and system context. *A blueprint for a quality information system must explain whose expectations will be met, describe those expectations and their relationships in a measurable way, indicate to certain extent what is required from system constituents and context elements to realize the expectations, and optionally suggest a procedure for verifying with respect to an intermediate or final system product that the expectations really are met.* The information system quality meta-model solves this problem by allowing for both multiple perspectives and an intersection of different models.

7.2 Constructs and definitions

In the final meta-model each construct is attached with a definition-assumption. This resembles the way how Wand and Wang (1996) present their theoretical model about data quality according to Järvinen (2001, 22-27). Wand and Wang keep definitions and assumptions apart, but in the quality meta-model they are merged together. Generally, definitions take the form "X means Y", "Y is called X", "X comprises" or like. Other propositions can be understood as theoretical assumptions. The account of model elements may also include some prescriptive propositions and practical implications or pose questions for further research.

Actor

Definition-assumption: People or groups of people who have some meaningful relationship with the information system under scrutiny are called *actors*. They are affected by the qualities of the information system or its products. A general naming for actors that is also used in the study is "stakeholders". Some actors may never use the system, but are anyway somehow interested in it or affected by it and its products. The term *informants*, in turn, means a sub-set of actors who actually participate in quality modelling or quality measuring and give some relevant information. It is the most important subset with respect to quality modelling. Actors are part of the information system context. They see the system and its environment from different perspectives according to their backgrounds and roles, and can possess different internal or external views of the system and its context. Perspectives and views affect the attribute definitions, prioritizations, goals settings and quality measurements made by the actors. Each quality model element can be traced back to a particular actor or actor group. Defining relevant actors and actor groups is actually *part of creating an activity model* for the quality modelling process and eventually *for the whole development process*.

All understanding of system qualities comes in the form of peoples' opinions and knowledge. Therefore the first step in quality modelling is mapping out, at least initially, relevant actors and informants. The broader notions, 'actor' and 'informant', compared for example to the traditional developer-user pair, enable to differentiate between the several roles people can have in relationship to information systems. They range from end users to investors, from developers to resellers, and so on. A very important actor role is the author or editor of quality model. Actors can even be collective, like organizational bodies and other groups of people. According to the definition above, an actor becomes an informant, if he or she is acceptable for giving some relevant information concerning the quality of the system. Information can also come indirectly from actors in the form of theories, scientific writings, rules of the organization, etc.

Traditional quality models did not pay enough attention to the variety of stakeholders. The big range of informants and their importance is well noted, however, e.g. by Siakas, Berki, Georgiadou and Sadler (1997), Georgiadou (2003), Berki et al. (2004), Siakas and Georgiadou (2005) and recently by Lagsten (2011). Siakas et al. (1997) present an actor categorization based on ISO-12207 (2008). Georgiadou (2003) lists following roles: acquirer, supplier, manager, operator, user, developer, maintainer and support process employer. Scholl et al. (2011) call different actor groups or archetypes of human actors "personae". Lagsten (2011) presents in her paper a method for evaluating information systems according to stakeholders. The principles of this method underline identifying and engaging all stakeholders, learning through the process, connecting evaluation to what actors do in practice with the information system, setting goals and evaluation criteria and finally acting to create change. Object oriented development method included even other information systems into the concept of 'user' (Jacobson et al. 1999, 5) and accordingly to the concept of 'actor'. In the quality meta-model other information systems are viewed as elements of the context and they can be discussed for example in connection with expectations about application integration.

Prescriptive proposition: The actor-informants are properly selected if all essential actor-stakeholder categories are represented.

Practical implication: Initial actor-informants must be selected as the first step in quality modelling and complemented with additional ones during the process according to needs.

Future research questions: How to map out all relevant stakeholder categories? Which actors are most affected by the information system and how? Who can give the most relevant information concerning the quality attributes?

Perspective and view

Definition-assumption: Human actors have certain perspectives on and views of information system and its context that are reflected in resulting quality models. A *perspective* is characterized by the actor's background, organizational roles and activities, beliefs, values, etc., and actor's relationship to the information system on basis of them. The former are at the same time elements of the information system context. A *view* of information system and its environment, in turn, is characterized by what it excludes and what it includes, i.e. by the constituents or elements visible in the view and their relationships. The elements in a view can be activities and processes as well as other things. A view of system can be concrete (through using or creating the system) or based on system descriptions. It can be general or detailed, partial or complete and so on. A view can be affected by factors constituting the perspective or other psychological and cognitive factors that determine what an actor wants to see or how an actor interprets what is seen. In terms of quality modelling a particular view usually covers only some parts of the information system and context.

For each actor, the knowledge about context, the knowledge about information systems in general and about the specific system under modelling together with the perspectives given by the roles and work the actor participates, constitute a kind of *frame of reference* that determines the appearance of the system to the actor in question (cf. discussion in section Metrics and Goal below). Perspectives and views which are presented and shared in the course of quality modelling form the intersection of individual frames. FIGURE 54 depicts the main elements of the frame of reference. The view of the system itself is the same simple structural-behavioural view that was used in Chapter 4 in connection with the presentation of initial meta-model. T1, etc. stand for "things" and S1, etc. for "systems" in the context of system and actor.

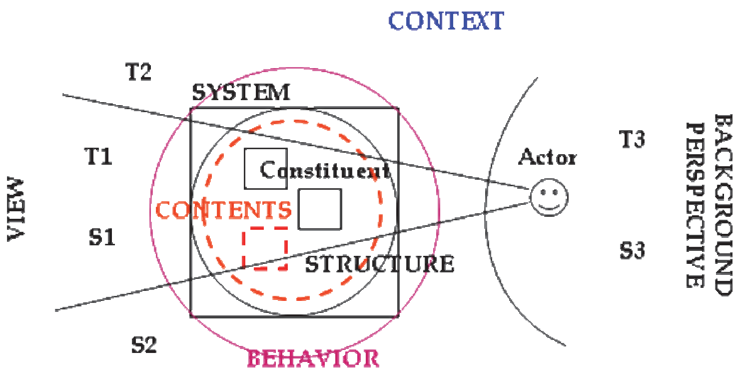


FIGURE 54 Main elements of actor's frame of reference

McCall et al. (1977) note three different orientations (operation, revision and transition) one can take when looking at a software product. These can be interpreted as actor's roles or activities, i.e. determinants of perspective. ISO 9126 document uses the term "perspective" to refer to activities like "acquisition, requirements, development, use, evaluation, support, maintenance, quality assurance and audit of software". Both the orientations and activities listed above are related to actors playing certain roles. Siakas et al. (1997) use the term "view" for perspectives. Examples are the contract view hold by acquirers and suppliers, operating view hold by operators and users, etc. Different views can be in agreement or conflict. Kotonya and Sommerville (1996), in turn, call perspectives "viewpoints" and introduce a requirements method with name viewpoint-oriented requirements definition (VORD). They map, however, viewpoints in addition to human stakeholders also to other systems interfaced to the system under modelling (cf. Actor construct in Jacobson et al. 1999, 5, 15). In the VORD model system provides services to viewpoints and viewpoints pass control information to the system. Wong and Jeffery (2002) base their software quality evaluation framework on the fact that evaluators are influenced by their job roles. The importance of perspectives in the same sense has been taken up later for example by Özkan (2006).

Research on user satisfaction and technology acceptance model (TAM) suggest still a couple of elements that constitute part of the actor's perspective, among them beliefs and attitudes about information system (see e.g. Wixom & Todd (2005)). Wixom and Todd (2005) create their own model for explaining how the usage of system is driven by actor's behavioural attitude toward its use and usefulness. The latter are assessments of the consequences of using the system to accomplish some task. Actor's behavioural attitude is according to the model influenced by beliefs about usefulness and ease of use. These, in turn, are based on attitudes and beliefs about information and system quality and satisfaction. At the beginning of the chain are then quality factors like reliability, flexibility, accessibility, etc. Kujala and Väänänen-Vainio-Mattila (2008), in turn, discuss how user's needs and values affect perception of information system. They also list activities and methods for including values in the system design process. And finally, determinants of actor's perspective can be sought from ethnic and cultural background. This is done for example by Srinivasan (2007), who presents findings in connection with designing an information system for 19 Native American reservations of San Diego County. Expanding TAM with group, cultural and social aspects has been suggested by Bagozzi (2007, 247-248).

Perspectives have a close relationship to areas of concern (domains) in the meta-model. It is obvious that actors in different roles have different concerns with respect to information system. This can be clearly seen in the discussion about two quality models by Özkan (2006). First of them, SOLE (Software Library Evolution) model (originally presented by Eriksson & Törn (1991)) categorizes quality factors into three groups according to actor perspectives. First category is 'business quality' (management perspective) and the concern is costs compared to benefits. Second category is 'use quality' (user perspective) and the concerns are what the system does for the user and how the interface is designed. Finally, third category 'IS work quality' (IS personnel perspective) deals with management, development, maintenance and operation of the system. These concerns can be compared, for example, to domains 'consumption of resources', 'support to business processes' and 'installation operation maintenance' used in the third case study.

For inexperienced informants it is extremely difficult to find, categorize and define quality attributes from scratch. Experimenting with different views can help a lot. The simplest view is the traditional black box view, where the information system or component under scrutiny is totally opaque and only inputs and outputs are visible. Any relevant elements from environment can be added to this black box view: for example, system in connection with related business process. With respect to each view the transparency of the system and the amount of details can be increased. In general, in the course of quality modelling the views of each actor are gradually being enhanced.

Future research questions: Are some views and perspectives more informative or productive than others with respect to quality modelling? Could an explicit sub-model of frame of reference be useful?

Information system

Definition-assumption: An information system as a technical artefact consists of electronic and non-electronic components (*constituents*), their functioning (*behaviour*) and relationships (*structure*). It includes the human and machine interfaces for data input and output. It is used to store, process, produce and present information (*contents*) in order to support human activities, including entertainment. Information products created by the system or serving as input into system are regarded as elements in a wider integrated system or the human information behaviour as a whole. From the perspective of product quality modelling the systems elements have no advance justification. The reason or explanation for their existence comes through their ability to contribute in fulfilling quality requirements.

The definition of information system differs, because of the definition of quality adopted in this study (see Section 7.3 below), for example from Alter's (2008, 451) "useful" view of information system as a work system. The latter includes human participants, whereas the meta-model treats them as part of context. According to the meta-model system-participant relationship, in other words what happens between the system and human participants, determines for its part the quality of the system itself. Alter (2008, 453) uses the word "technology" for the hardware and software and calls it a "tool view".

The first two sentences of definition-assumption correspond to what is called internal view by Wand and Wang (1996) and the rest to what is called external view. In connection with quality modelling the initial view is external and at first the information system is seen as a mere black-box. The purpose of this is to clearly discern the desired relationships between the system and its context and only after that give justification to the internal system constituents on basis of those relationships. Gradually, according to the needs of attributing some qualities to lower level components and finding contributors to the qualities from inside information system, more detailed architectural descriptions are used. Good architectural descriptions are worth of gold but unfortunately a rarity in system development projects. In addition to the articles about architecture analysis referred earlier, a good treatise of the subject if given by Bass, Clements and Kazman (2003). Developers are the most important actors in respect of system descriptions.

Context

Definition-assumption: From the perspective of quality modelling *context* comprises all the elements in system environment that are members in the desired relationships between the system and context. These relevant entities can be human or non-human, independent of the spatial or temporal distance. Context is an essential sub-model in the quality meta-model.

Some important sub-areas of context include human and technical infrastructure, business processes that the information system has to support, related information systems, actors and actor groups using the system. A special part of context for an electronic information system is the wider immediate context of information processing including human mind, cognitive processes, language and traditional manual tools. In areas like Africa even climate must be taken into account. In different phases of quality modelling and in connection with different quality attributes only parts of environment can be set in focus. *An appropriate meta-model of context could help in identifying and handling the relevant contextual entities*. An idea of "context domain model" has been recently taken up by Jarke, Loucopoulos, Lyytinen and Mylopoulos (2011). General guidelines can be sought for example starting from Alter's (1999, 2008) work system theory. The importance of context has been underlined in many writings about quality and requirements engineering (e.g. Kotonya & Sommerville (1996), Özkan 2006, van Reijswoud 2009 (with respect to developing countries), Ali, Dalpiaz & Giorgini (2010)). Van Reijswoud (2009) lists five different aspects of context relevant in developing countries: culture, physical conditions, organizational structures, economy and political climate. Holmström and Sawyer (2011) take up an important point about the actor organizations as part of context, namely that organizations using information systems do not remain unchanged even during the period starting from requirements gathering and ending when the system is developed. This and other challenges caused by change must be addressed through iterative and continuous requirements process.

Prescriptive proposition: A proper description of context accounts for all the elements in system environment that relate to the qualities included in the overall quality design.

Future research questions: What are the elements of a context model that best fits the needs of quality modelling? How does the physical, infrastructural and cultural context reflect in quality models? The latter question implies joint research projects carried out, for example, by social and computer scientists.

Quality attribute and attribution

Definition-assumption: A *quality attribute* refers in first place to a desired relationship between information system and its context or to a number of connected relationships. The existence and intensity of this relationship determines the level of quality in question. In a domain-attribute set each quality attribute can be given a general *definition* and a *goal value*, and it can be *attributed* or allocated to the system as a whole or some of its constituents. Having goals is essential in defining and assessing quality. Attribution to a constituent means that the particular quality of the constituent determines in fact the same quality of the whole system. Qualities are of high importance to actors and form a prioritized set.

By taking ISO 9126 (2001) quality model's six main characteristics functionality, reliability, usability, efficiency, maintainability and portability as examples, one can see that they all refer to a relationship between information system and its context. Functionality, that corresponds to usefulness and support to business processes in the case studies, is the capability of software to provide functions which meet stated and implied needs (ISO 9126 2001, 7). It is clearly a relationship between the product, its users and business processes they have to carry out. Reliability is defined as the capability of software to maintain a specified level of performance under specified conditions (ISO 9126 2001, 8). Again it is about a relationship, this time a relationship between the product, a specific instance of using it and certain conditions. Usability is the capability of software to be understood, learned, used and attractive to the user when used under specified conditions (ISO 9126 2001, 9). This characteristic relates the product to the user and certain conditions. Efficiency is the capability of software to provide appropriate performance, relative to resources used and under stated conditions (ISO 9126 2001, 10). It relates the product to resources and use under certain conditions. Maintainability is the capability of product to be modified and adapt to changes in environment, requirements and functional specifications. Again it is clearly about the product in relation to its context. Finally portability is defined as the capability of software to be transferred from one environment to another (ISO 9126 2001, 11). Similarly *most in the wide range of* new *quality attributes* (not included into ISO 9126 2001) *refer to some kind of relationship between system and its context*. Another comparison can be made to the conceptual framework used in unified software development method created by Jacobson et al. (1999). The authors divide all requirements into two main categories, functional and non-functional. The former group corresponds to ISO 9126 (2001) model's first main characteristic 'functionality', and the latter broadly to the remaining characteristics. Consequently all requirements in general can be, in the end, traced back to requirements about the relationships between information system and its context.

In accordance with the above definition and reflection, the main difference between *quality attributes* and *other attributes*, or quality requirements and other requirements, is that *the former refer in first place to desired relationships between information system and its context and have more importance to actors than the latter,* and *the latter can often be derived from the former*. In addition following distinctive features were attributed to quality requirements in Section 6.4:

- Actors set usually goals regarding to what degree quality requirements need to be met.
- Priority, definition and measured level of implemented quality requirements can be relative to use case or scenario, actor, or some other feature of the context.

As was stated in connection with the initial meta-model, quality attributes and requirements have been given different names, like "factors", "concerns", etc. Still other valid names are "criteria" and "values" (Scholl et al. 2011, 790). How the desired relationships or state of affairs between system and context is achieved is another important aspect in quality models that is discussed below under heading contributor.

Domain, domain-attribute set and priority

Definition-assumption: *Domain* is a field of thought or area of concern to the actors in connection with which a quality attribute or group of attributes is relevant. It groups together related attributes that can be viewed as individual concerns. Each domain in an attribute set or collection is given a general *definition*.

Definition-assumption: A *domain-attribute set* is a *prioritized* list of all quality attributes ascribed to the information system as a whole or to a particular constituent. The attributes in the set are given a general definition and goal value, grouped according to prioritized domains and related to each other. Attributes themselves are prioritized on the level of the whole set, inside domains or both. Priority is characteristic of quality determining attributes. Different factors, among them actors' perspectives and views, have an impact on the final selection and prioritization of attributes. Quality *domain-attribute collection*, in turn, is a general set or supply of domains and quality attributes that can be used as a source when assembling the system specific domain-attribute set. Specific attribute collections can be created for different types of systems.

Definition-assumption: According to discussion about quality attributes above, quality requirements are, as such, more important to actors than requirements in general. *Priority* is one of the factors that make the existence or degree of something to be a quality in the eyes of the beholders. Priority inside quality model refers to the ranking of quality attributes and domains in relation to each other. Prioritizing attributes and domains can be logically differentiated from prioritizing system constituents, structures or behaviour. One can rank qualities just on the level of entire system without yet attributing them to a specific constituent.

Based on the discussion above about nature of quality attributes and quality requirements compared to requirements in general, one can say that the overall *quality design embodied in the domain-attribute set is the actual master plan or "DNA" of the information system*. In the account of case studies it was called "attribute set", but a more descriptive name is domain-attribute set. A *domain-attribute set is in principle open to new domains and attributes and to regrouping or even removing of attributes*. Obviously the total number and selection of good characteristics required of a particular information system varies case by case and from time to time. In fact, even during one and same quality modelling process the domain-attribute set can change, and the early categorization differs from the final one. Pre-collected general or system type specific domain-attribute collections are needed to assist in quality modelling. They act as starting points and checklists ensuring that the experiences of similar information systems and similar environments, or information systems in general, are taken into account. Possibilities for generalization and specialization are endless. A plain and general list of quality attributes is presented for example by Siakas et al. (1997). It is ordered alphabetically and called "attribute alphabet". Nelson et al. (2005, 207) present a two-part list of attributes, one for information quality and one for system quality. System type or business specific collections are sometimes called domain-oriented models (e.g. Villalba et al. (2010)). An example of system type specific list is the one for B2B applications constructed by Behkamal, Kahani and Akbari (2009). Another one is quality model for security products in Villalba et al. (2010).

A bulky domain-attribute set assigned to a large information system is not easily comprehensible to informants. Therefore, it makes sense *to split it into a number of smaller sets* each describing a part or aspect of the entire system. Several alternatives exist: differentiating between hardware and software qualities, between subsystems and components, etc. Attribute sets can also be domain specific listings, for example, only the qualities required from data and contents or user interface.

Domain and quality attribute set with priorities and attribute relationships *can be seen as a major sub-model within the entire quality meta-model*. It is commonly used in a more or less complete form in studies about software quality. Siakas et al. (1997), for example, present in addition to the plain list mentioned above a general attribute set under the name "software quality metrics model". It combines quality attributes combined with other measurable attributes called "criteria". General attributes are arranged into two groups, pertaining to product operation and revision respectively.

Future research questions: What are meaningful ways of categorizing quality attributes? What is actually common to a set of attributes? Pre-created system type specific attribute collections? General quality attribute set for environments like East-Africa? How do prioritizations made by actors relate to context? How does informants' view of overall quality design change during development process?

Use case and scenario

Definition-assumption: *Business use cases*, as part of the business model, represent the processes of an organization with or without explicit reference to supporting information systems. A *system use case*, for its part, represents the use of system per se, without, in the first place, drawing attention to its connection to business processes. The term *scenario*, in turn, refers to particular circumstances or to a flow of events, other than use cases, where the role of human actors as users of an information system sometimes can be non-existent, or not focused on.

Use cases and scenarios constitute contexts in which actors can find detailed definitions for quality attributes and in which existence or lack of qualities can be noticed and measured. These contexts are indispensable for quality modelling. Differentiating between scenarios and use case types helps to understand quality requirements more deeply and accurately. Many actors are not however, familiar with this kind of business modelling terminology and need guidance. The importance of use cases is noted often in literature. One recent example is Lagsten (2011) where the activities what actors do while using the information system is taken as a starting point for evaluating the system itself. “Scenario” term, in turn, is used in Scholl et al. (2011) for archetypes of human action and information use environments. It seems to cover both business and system use cases and scenarios defined above. Still another example of importance of business processes for requirements engineering is the article by Jarke et al. (2011) about new challenges facing requirements engineering. Business process is raised as a key unit of requirements analysis.

System-context relationship, indicator and model

Definition-assumption: Most of the characteristics that are traditionally called ‘quality attributes’ refer to a desired *relationship* or set of desired relationships *between the information system*, or its constituent, *and* one or more entities in the *context*. This can be regarded as characteristic of quality attributes. The relationship can also be named a desired state of affairs. In this way the system is linked with context entities in actors’ minds or physically. Certain facts, called *indicators*, indicate the existence and intensity of the relationship in connection with real use cases and scenarios. A quality model can in addition list negative facts that show the lack of the desired relationship. A general formal presentation of the desired relationship is called *model*. It can be given, for example, in the form of an entity relationship (ER) model.

The term “predicate” was used in the definition element of initial meta-model. It was taken from the vocabulary of logic. In final meta-model the object properties and polyadic relationships can be described with the model element. The *desired relationship(s) between information system and its context is the essence of quality and quality requirements*. It can also be called desired state of affairs. And the *designing these relationships is the ultimate goal of quality modelling* and in fact whole system development. All other meta-model elements have a subordinate status and help to define, explain and implement the desired relationships.

Indicators are a valuable part of attribute model. They *make qualities tangible to actors and form the basis for metrics*. There are many counterparts in the literature for indicators. The questions, for example, developed by Boehm et al. (1978) for judging the quality of software product can be converted into positive assertions that correspond indicators in quality meta-model. E.g. question about is the code free of obvious errors can be converted to statement that the code is free of obvious errors. Some writings, like Niemelä and Immonen (2007) use the concept of ‘goal’ in a similar meaning, as well as Chung et al. (1999). Niemelä and Immonen (2007) have adopted a good practice of attaching a rationale to each goal. Folmer and Bosch (2007), in turn, use the term “indicator”.

Future research questions: How explicitly and well do publicly available quality models define desired relationships between system and context?

Contributor

Definition-assumption: *Contributor* is a thing inside or outside the information system that affects in a positive way a desired relationship between system and its context. It can be alternatively called factor. The description and understanding of system environment, discussed above, is important in identifying the external contributors. The internal things, in turn, are usually system constituents, behaviour (functioning) or structures and consequently part of the system model. Desired relationships are the raison d'être of contributors and quality attributes can be said to refer in the second place to the latter. What are the contributors in reality with respect to each quality is in the end a subject of empirical study. Thereafter, based on achieved theory system developers can instantiate the contributor elements in system design.

It is important to note that sometimes *different contributors can realize the same quality to actors*. More or less different program code, for example, can bring users the same experience of usefulness. On the other hand, *different actors can experience different degree of quality regardless of exactly the same setting of contributors*. Therefore, according to the meta-model, quality attributes refer in first place to the desired relationships between system and context and then, in second place, to the contributors. Accordingly, as was observed already in the first two case studies, the essence of a specific quality to actors can be basically described with two meta-model elements in following order: 1) desired relationships between the information system and its context and 2) contributors.

Like for indicators, there are many counterparts for internal contributors in the literature. They are called "internal" or "tangible" properties by some researchers (e.g. Ortega et al. 2003, Dromey 1996). McCall et al. (1977) define software quality factors, like efficiency, using constructs which they call criteria, like execution efficiency for efficiency. Criteria, in turn, are defined as "attributes of software". For execution efficiency they are "those attributes of software that provide for minimum processing time" (McCall et al. 1977, section 4, page 5). Chung et al. (1999) use the term "operationalization" for internal contributors. Operationalizations are development techniques (methods) for accomplishing softgoals, i.e. quality attributes.

As can be seen from the discussion in Section 2.2.9 about architecture analysis, one of the contributors that have got a lot of attention is the structure of information system. It is actually natural that the search for internal contributors starts from top down, i.e. from development platform and system architecture. Later Niemelä and Immonen (2007) have been working on this architecture line and presented a method called "Quality Requirements of a Software Family (QRF)" for defining quality requirements and converting them to architecture. In defining the requirements the method notices different actor categories starting from business experts who know the product market. Another recent example is Folmer and Bosch (2007). They present a framework for connecting software architectures to qualities (based on ISO 9126 2001 model).

Narasimhaiah and Lin (2010), in turn, have studied external contributors. They focus on organizational and individual human factors associated with quality attributes like reliability, ease-of-use, maintainability, usefulness and relevance. They found as external determinants of software quality things like attitude of management responsiveness and capability of IS department and capability of users themselves. Each of the determinants was further decomposed into smaller items. Capability of users, for example, consisted of users' knowledge in the system, training received, involvement in or resistance to the system, and technical competency. When it comes to super-attributes like sustainability the importance of external contributors is evident. This has been discussed especially in articles about ICT for development. One such article is Silva (2007). He takes up land administrations systems of which the context of third case study is an example. Among obstacles to successful implementations are mentioned even such things as institutional jealousy and historical resistance to re-examine the institution of land ownership. Silva (2007) uses the term "institutionalization" to refer to the adoption of systems by organizations. From the viewpoint of the study at hand it is an element of sustainability. Another good listing of external factors influencing information system sustainability in least developed countries has been given by Kelegai and Middleton (2004). They studied organizations in Papua New Guinea.

Wagner (2010) advocates so called activity based quality models (ABQMs). One of the basic constructs is 'entity' which comes very close to the meta-model element contributor. For Wagner an entity can be any thing that can have influence on software quality (Wagner 2010, 1231). Entities possess attributes and an entity-attribute combination is called fact. Facts can be assessed by different measurements. Facts have positive or negative impact on activities. Maintenance and modification are examples of activities, and maintainability and modifiability corresponding qualities. Both facts and activities are organized in hierarchies. High-level activities, for example, can be divided into sub-activities. Activities are comparable to use cases and scenarios in the meta-model. Other similarities include that ABQM has as top level entities 'system' and 'environment'. It puts, however, 'organization' on the same level as 'environment'. The meta-model, in turn, views organization as a part of context (environment). To assess and predict the quality of software more precisely on basis of ABQMs Wagner (2010) uses a Bayesian network between facts, activities and indicators. Indicators are metrics for facts and activities. A lot of empirical knowledge about the dependencies is a prerequisite for building a valid quantified Bayesian network.

Future research questions: Finding internal and external contributors to the growing set of quality attributes is a never ending challenge to empirical research.

Attribute relationship

Definition-assumption: *Attribute relationships* are a feature of domain-attribute set. These relationships can be identified by comparing indicators, models, contributors and measured values. Indicator sets, for example, can be separate, overlapping, or one included in another. Contributors, in turn, can be indifferent to one another, cooperating (supporting), or conflicting. In theory *there can be as many kinds of attribute relationships as there are relationship types between indicators or contributors*.

The term "connection" in the element name of the initial meta-model has been changed to "relationship" in the final meta-model. Compared to "relationship" "connection" might have a connotation of being a positive relationship. *Attribute relationships can be identified or "predicted" by comparing attribute models*. Full understanding of relationships, however, can be achieved only after a sufficient number of measurements.

Positive and negative impact (trade-off) was clearly noted already by McCall et al. (1977). Attribute relationships have received since then a good deal of attention in scientific papers on information system quality. One such line of research has been architecture analysis. It tries to determine which software architectures support certain quality attributes more than others. (E.g. Svahnberg & Wohlin 2005) The selection or composition of the architecture for a particular information system can be made on the basis of this knowledge. One of the basic notions within architecture analysis is that some quality attributes are *conflicting* and cause a trade-off situation, while some *support* each other. Cysneiros and Leite (2004), for their part, use the expression "*positive* or *negative contribution*". They seek out these relationships by systematically comparing graphs representing non-functional requirements. Earlier Chung et al. (1999) had developed a graphical technique called softgoal interdependency graphs (SIGs) for visualizing attribute (non-functional requirements) relationships. The graphs show also how softgoals are related to operationalizations (internal contributors in the quality meta-model). Recently López, Inostroza, Cisneros and Astudillo (2009) have developed a formal semantic web based technique of representing SIGs. It uses OWL (Web Ontology Language) and is called NFRs and Design Rationale (NDR). Siakas and Georgiadou (2005) identify *reducing-enhancing* relationships between quality attributes in connection with interest conflicts between different stakeholders. Egyed and Grünbacher (2004) relate attributes to requirements. Qualities can be *indifferent* in respect of one another, *cooperating*, or *conflicting*. If two requirements affect the same part of a system and their test scenarios execute the same or similar lines of code, there is the possibility that the requirements and corresponding attributes will either conflict or cooperate. The overlapping of the execution path of a test scenario is referred to as trace dependency. In addition to the above examples from literature, quality attributes can be compared by indicators or by other model elements. This way they are found to be *separate*, *overlapping*, or one *included in another*. Sustainability, for instance, is a clear example of a composite attribute, and it includes affordability, being well-supported, and so on. Khaddaj and Horgan (2005) use a chart to depict the attribute relationships. It is a matrix with columns and rows representing the same set of quality factors. In the intersections of rows and columns a symbol indicates the relationship of factors in question.

Existence of conflicting attributes and trade-off situations means that in reality all measured attribute values seldom are maximum possible values. Therefore, a system specific *attribute set must be finalized by analyzing attribute relationships and attaching goal values to attributes* that indicate a balance based on how conflicts have been solved, i.e. to which of the conflicting attributes have been given priority over others.

Metrics and goal

Definition-assumption: A complete arrangement for measuring the existence and degree of a quality, i.e. of a desired relationship between system and its context, includes selection of *instrument, unit, scale, actors, and measurement procedure*. The *object of measurement* is in first place the existence and degree of certain desired system-context relationship represented by the indicators, and in the second place the existence and properties of internal and external contributors. The objects of measurement are in each case defined, observed and measured in a certain *frame of reference*. The things that cause relativity, i.e. difference of results compared to another frame of reference, constitute together the set of axes of the frame. They include, for example, business and system use cases, business processes, observer's organization and role in it. Relativity is characteristic of information system quality.

Definition-assumption: *Goal* is the desired degree or intensity of the desired relationship. Desired values are indicated as part of general quality design in the system specific attribute set. Trade-off situations, for example, and the compromises made affect the setting of desired attribute values. In addition to a goal value quality attributes can have other values like *reference value* (averaged value in similar contexts), *maximum value* and *lowest acceptable value*.

Metrics has always been a part of quality modelling either in the form of measuring individual qualities or in the form of measuring the overall quality of the system. The terminology used for referring to different aspects of metrics has been varied. Nielsen (1993, 115), for example, discusses under title "usability heuristics" a method for inspecting the overall usability of user interface. It is based on judging if the interface complies with a set of usability principles. From the viewpoint of the proposed quality meta-model these principles correspond either to elementary quality attributes and their indicators or to contributors. The idea is to let selected evaluators to examine the user interface and produce a list of conflicts with the principles. Parts of the measurement procedure can also be automated. This has been in the minds of quality model developers from the beginning (e.g. McCall et al. (1977), Boehm et al. (1978)). Before the values of quality attributes can be measured respective requirements must be quantified, i.e. made measurable. Again, several methods exist. One often used is finding out measurable sub-characteristics for a higher level characteristic (noted e.g. by Glinz (2008)). The same method is used by Seffah, Kececi and Donoyaee (2001) in their framework for quantifying usability. Authors call measurable sub-factors criteria. In the proposed quality meta- model measurable indicators can play this role. The measuring arrangement has also been widely discussed. Boehm et al. (1978) use the term algorithm for measurement procedure. According to Glinz (2008) a measuring arrangement consists at least of a scale, procedure, lowest acceptable value and a planned value. Kitchenham and Pfleeger (1996) generalize that from user perspective measuring a quality feature includes directly measurable attribute components, unit, and tool, as well as planned level of quality. An extensive account on all aspects of metrics is given, for example, by Fenton and Pfleeger (1998). And if one wants to go deeper into the ontology of software measurement concepts and terminology a good presentation and proposal is given by García, Bertoa, Calero, Vallecillo, Ruíz, Piattini and Genero (2006). They prefer, for example, the term "measure" to "metric". And they introduce the term "indicator" meaning a measure that is derived from other measures. A measure is a measurement approach or method and the measurement scale. The quality meta-model sub-element indicator and indicator construct of García et al. (2006) meet in a sense that quality model indicators can often be converted into measures.

Sophisticated measuring arrangements do not necessarily shine extra light on quality. Simple instruments and rough scales on their own can often give the essential information. The unnecessary quantification is discussed for example by Glinz (2008). According to him for some requirements no metrics exist or measuring arrangement is too expensive. Each definition and measuring effort should be relative to the real value it delivers to stakeholders. Glinz develops a method of categorizing quality requirements by a set of factors that help to decide what kind of certification is needed or is it needed at all. Among the factors are criticality and distinctiveness of the requirement in question for the product, effort needed to quantify it, validity of obtained measurements, shared understanding about the requirement between stakeholders, required certification, etc. And sometimes one "needs to substitute human judgement for measurement" which "is not a priori bad and subjective" (Glinz 2008, 39).

Future research questions: What are the most important axes in the frame of reference for modelling and measuring quality? How does the measuring process refine the definition of quality relationships? How much do the measured values vary and according to what?

7.3 Concept of quality

Quality of information system as a technical artefact in its context of development and use is *in the first place* determined by the existence, lack, intensity or number of desired relationships between information system or system constituent and its context. These relationships have *priority* over other things. At the same time their importance and observed intensity of the actual relationships *can vary according to different factors*. Human actors and human work systems are the most important elements of context with respect to quality. The desired relationships are described in attribute models using indicator propositions and entity-relationship models. *In the second place* quality of information system is determined by the existence, lack, number of or intensity of elements, inside or outside the system, contributing to the realization of the desired relationships. Known quality expectations and needs are expressed as *quality requirements*. *Requirement* as such is a broader notion meaning anything required, be it a certain quality, or something needed to realize it, or something else needed in the system for some reason.

Overall quality of information system is determined by realization of overall quality design embodied in the system specific attribute set which attributes the qualities to the system as a whole or to individual constituents, defines priorities and balanced goal values and describes the relationships between attributes. Any formulae created to calculate or derive a value for overall quality must be capable of taking into account the priorities and goal values.

Basic English dictionaries (e.g. Cambridge Advanced Learner's Dictionary 2008, 1162) define quality basically as 'the good characteristics' which something has. And the etymologically related pair of Latin words "qualis" (what kind of) "qualitas" (quality, property) is no more mystical. Reeves and Bednar (1994) have discussed different definitions of quality in business context. They state that the concept of quality has had multiple and often unclear definitions and look more closely at four of them: quality as 'excellence', 'value', 'conformance to specifications', and 'meeting expectations'. Excellence means meeting the highest criteria in some area like intelligence, strength etc. The value aspect introduced price, or value, as an additional determinant of consumer's decision. Different compatibility requirements in production of component based machines lead to equalling quality with conformance to specifications and to making quality measurable. Finally the most pervasive definition 'meeting customer's expectations', according to Reeves and Bednar (1994), grew out of services marketing. It is also the most complex definition and most difficult to measure. Reeves and Bednar (1994) conclude that a global definition of quality does not exist and different definitions are appropriate in different contexts. The IEEE Standard Glossary of Software Terminology (1990, 60) offers a two part definition of system quality: "the degree to which a system, component or process meets specified requirements" and "the degree to which a system, component or process meets customer or user needs or expectations. It coincides with the fourth definition discussed by Reeves and Bednar (1994).

The study at hand started defining quality from the viewpoint that, in very general terms, information system quality is determined by the existence and intensity of something pertaining to the system, something identifiable and desired by actors and stakeholders. This point of view is in a way similar to the definition of quality as 'meeting expectations' above. What is desired is usually referenced to by using adjectives and abstract nouns and interpreted as *characteristics or features that reside inside and constitute an integral part of the entity* *(system) being described*. Quality definitions like in ISO 9126 (2011) reflect this viewpoint. It gives in the annex a general definition for quality taken from ISO 8402 1994 (replaced now by ISO 9000 2000): "the totality of characteristics of an entity that bear on its ability to satisfy stated and implied needs" (ISO 9126 (2001, 20)). Analysis of the meaning of *adjectives and abstract nouns used for qualities*, like "usefulness", discloses, however, that they *refer actually to certain relationships between the system and things in its context*. A proposition like "Statisticians can produce all needed statistical presentations using the EMIS" (a definition element of usefulness in EMIS case study), for example, refers to a relationship between the EMIS, statisticians and particular business tasks which, in turn, is part of some larger business process. The discussion above in connection with quality attribute element, in turn, showed how all ISO 9126 (2001) quality model's six main characteristics in fact also refer to system-context relationships. Consequently, most of the requirements that are understood as quality requirements can be assumed to pertain to relationships between the system and its environment, either between the system and user or some other things in the context or both. The existence and intensity of these relationships is then determined and affected as well by internal system constituents as external things.

The search for the essence of quality can as well be started from inside the system. In the case studies the initial view of information system was as much as possible the structural-behavioral view. This view ignores in principle the context. In accordance with the *structural-behavioural view* any information system can be described exactly and without remnants by indicating its architectural type, programming language used, design patterns, layers and packages, by listing procedures and methods, records in the database, series of instructions, and so on. This kind of description *does not*, however, *explain WHY these constituents are required or desired*. According to the proposed meta-model the explanation lies in the way these things affect the desired relationships between the system and its context, i.e. qualities, hence the name "contributor" for these internals. In some cases an element is needed to create another element. But for what is the latter needed? Every chain of *WHY*s leads in the end out of the system internals into some relationship between system and its context, even if it is just a relationship between system and individual actor. Accordingly, to explain *WHY* a design pattern, method, etc. is required or needed, these relationships must be described, understood and related to the elements of structural-behavioural view. In a way, by specifying quality requirements we actually position the information system in a particular context.

FIGURE 55 depicts the above viewpoints. It lists along the upper part of the ellipse a number of terms used in literature for referring to those subcategories of requirements that are commonly regarded as quality requirements. Functional requirements are added to the set on basis of the discussion earlier in this study. Each individual requirement (NR1, G1 and FR1) or "desire" points to a relationship (R1, R2 and R3) between information system and its context. At the same time these requirements point to certain concrete system features (e.g. architecture and method) and things (T1, T2 and T3) outside the system. The former, quality requirements, explain the need for the latter, concrete system features and external conditions. Accordingly, if a developer starts asking in respect of any internal system feature that is under design or implementation, why is it required, he or she finally always traces back to some desired relationship between the system and its environment. This subordinate status of system internals and externals compared to the expected system-context relationships explains the meaning of the expressions "in the first place" and "in the second place" in the definition of quality above.

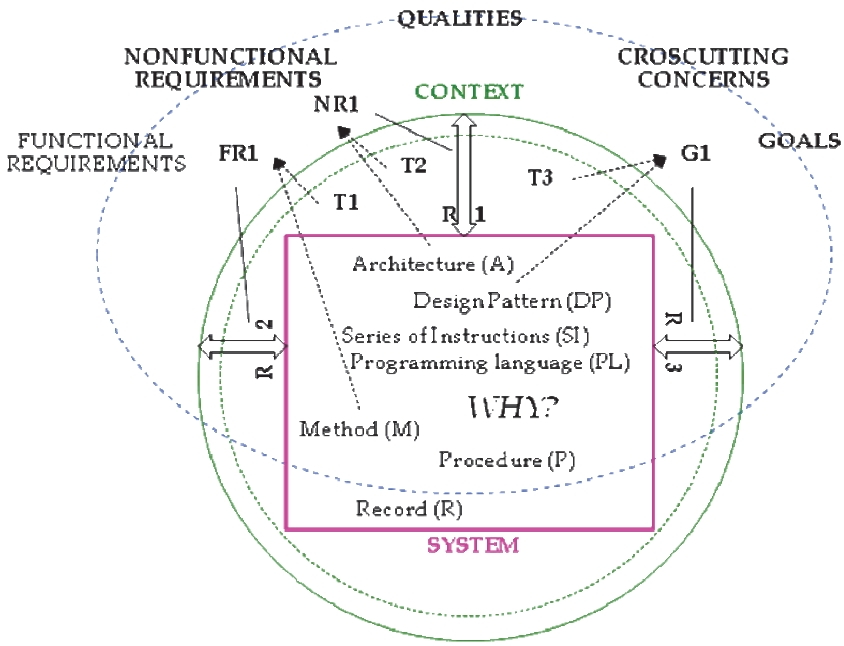


FIGURE 55 Quality requirements, context, internal system features and external contributors

The above viewpoint can be even widened. In connection with the discussion about models and modelling, in the beginning of Section 2.1, the use of information was put in the context of human societies, organizations, personal lives etc. Firstly, whenever actors create, gather, store and process information it is done in order to serve the goals of human activity. Secondly, since first ideographic scripts almost 10 000 years ago, different tools have been used in dealing with information and the quality of information tools have always been determined by their added-value in this process. Consequently, the system context relationship is in fact two-tiered and there are *two quality determining interfaces and relationships*: 1) *between information system and information processing* in general (the inner dotted circle in FIGURE 55) and 2) *between information processing and other human activities* (included into the rest of context). Both of these contain the raisons d'être even for contemporary electronic information systems and their constituents.

The importance of system-context relationships to quality considerations is clearly already in the historical work of Victor R. Basili. He developed, for example, a quality measurement model called Goal Question Metric (GQM) (Basili (1993), Basili, Caldiera & Rombach (1994), Shull, Seaman & Zelkowitz (2006)). It relates the information system and its quality to the goals of organization, up to the level of policy and strategy. The understanding of quality as a relationship, where concrete and as such neutral system features play their role, is also related to Kitchenham and Pfleeger's (1996) user perspective according to which quality equals to the appropriateness of the product for a given context. It looms also out of Ortega's statement that "fundamental axiom of product quality is that the tangible internal characteristics or properties of a product determine its external quality attributes" (Ortega et al. 2003, 220). Similarly it is behind the differentiation between internal, external and quality in use attributes in ISO 9126 (2001) standard. Internal attributes influence external attributes and the latter influence quality in use attributes (ISO 9126-1 (2001, 3)). At the end of the chain quality in use refers to the capability of software to enable users to achieve specified goals in a specified context of use. All this fits perfectly the characteristic that differentiates according to Gregor (2006) information systems from other fields of study, namely that it examines a technological system and social system together and the phenomena that emerge when the two interact. It complies also with the new challenges faced by requirements engineering discussed by Jarke et al. (2011). One of the new principles is called "intertwine requirements and contexts" (Jarke et al. 2011, 997-998, 1003). The authors see requirements and contexts unavoidably intertwined and requirements as "boundary objects" in the intersection of technical and social domains. Designers have to monitor changing world and sustain an adequate correspondence between system and context.

Glinz (2008) tries to relate quality requirements to requirements in general. He calls quality requirements "attributes" and differentiates them from functional requirements and constraints. The latter can be physical, legal, cultural, etc. Quality requirements, in turn, pertain to performance or are specific "-ilities" like reliability, usability, security, etc. This study takes another viewpoint. All prioritized expectations about the relationship between the system and context are quality requirements. These expectations can be further categorized, related to each other and analyzed according to principles and structures offered by the quality meta-model.

Voas and Agresti (2004) pay attention to the difficult question of overall quality and how to measure it. They presume that product quality comprises a set of key attributes, like reliability (R), performance (P), fault tolerance (F) etc. Accordingly, overall quality is a function of these attributes plus an error component representing aspects that the key attributes cannot define. The authors admit that the formula is simplistic and inaccurate, but fits to the realities of practical system development. Voas and Agresti (2004) suggest further that when calculating the total quality individual attribute values should be normalized. For each attribute the highest value is 1 and it corresponds to the maximum achievable level of that attribute. Finally, the attributes are weighed according to their importance. All this is in accordance with the understanding of overall quality presented above in the definition-assumption. FIGURE 56 depicts the meta-model elements that constitute the overall quality design. These are the prioritized domain and attribute sets with attached definitions. Attributes are related to each other and given goal values. Attribution indicates if the quality in question is required rather from a specific system constituent than of the entire system. Use case and scenario element are needed if a quality definition and goal is valid only with respect to a particular use case.

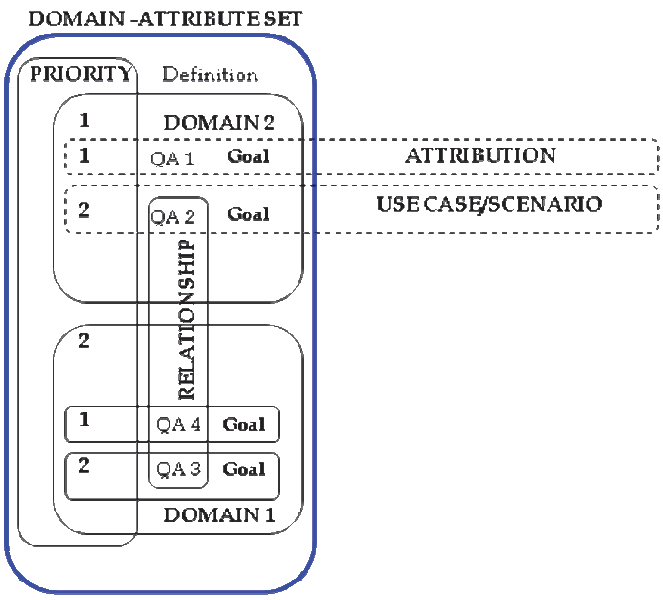


FIGURE 56 Elements of overall quality design

7.4 System specific quality models

System specific quality models, i.e. quality models created using the meta-model as a template in actual system development projects, can be more or less rigorous. Some of them are just prototypes as those created in the three cases presented in Chapter 5, and some of them are detailed and elaborate, all depending on the needs and available resources. The most important thing is that in each project there exists a quality model which leads and gives direction to the development process. It is necessary as well that quality attributes are attached with tangible indicators and that the degree of quality is measured according to them. Finally it is important to be aware of the taste of subjectivity and relativity inherent in quality models.

A system specific quality model can be compared to a multidimensional database or an OLAP cube. Each cell is a conjunction of dimensions. The essential dimensions in quality meta-model are quality attribute, related attribute and relationship type, domain, system constituent, use case or scenario, related context element and relationship type, actor, perspective and view, goal value and priority. Each of these dimensions is a non-overlapping hierarchy. If the data contained in the system specific quality model is entered into a database, it can be analysed like any multidimensional database. It can be queried by actor, by attribute, by domain etc. One can find out how an individual actor looks at the information system from a certain perspective seeing certain parts of the system and its context, and having certain use case or scenario in mind. This kind of traceability is an important feature of models, noted very well for example by Niemelä and Immonen (2007, 111) in their model for software family quality requirements.

7.5 Evaluation

This section compares the proposed meta-model to some newer theories that were referred to as alternatives or competitors in Section 3.1 above. The writings are presented in a rough chronological order. First of them, Ortega et al. (2003) builds on Callaos and Callaos (1996) idea of systemic quality that consists of product efficiency, product effectiveness, process efficiency and process effectiveness. These four top level model elements are called dimensions. In other words, the model takes into account both development process and the end product. Another example of adopting systemic quality model is given by Rincon, Alvarez, Perez and Hernandez (2005). Under dimensions lie elements that are called categories: on product side ISO 9126 (2001) model's six main quality characteristics and on process side five process categories. The latter are client-supplier, engineering, support, management and organizational processes. Each category, in turn, has a set of associated characteristics that must be fulfilled in order to control product or process quality. Finally each characteristic has a group of metrics. In their case study Ortega et al. (2003) derive from ISO 15504-2 (1998) metrics for characteristics and prove how, according to measurements, process efficiency and effectiveness influence product quality. Rincon et al. (2005) compared in their study four discrete-event simulation applications intended for Venezuelan oil industry. They did not, however, implement the process sub-model.

Compared to systemic quality model this study takes the view that it is extremely important to create a model for system development process, and it is as well important to have a theory about the relationships between process and the end product. It is however not necessary and can even be confusing to combine all this in one and same model. The systemic quality model is in addition missing many of the aspects of product quality that are included in the quality meta-model, among them system context, human actors and their perspectives, prioritization etc.

Georgiadou (2003) titles her article "GEQUAMO - A Generic, Multilayered, Customizable, Software Quality Model". According to the title one expects to step on the meta-level of quality modelling. The article discusses selected aspects of quality: stakeholders and their viewpoints, multilayered decomposition of attribute into sub-attributes and metrics. Attribute relationships are also shortly touched. Generality and customizability seem to mean that the set of attributes constituting system's quality profile can vary according to project and stakeholder. In addition Georgiadou presents diagramming techniques for visualizing quality profiles with quantified attribute values. According to the paper quality is "the degree to which software meets customer or user needs or expectations", and "a quality factor is an attribute of software that contributes to its quality" (Georgiadou 2003, 317).

Compared to Georgiadou (2003) the proposed quality meta-model covers more aspects of product quality, among them prioritization, domains, use cases and scenarios, etc. Otherwise human actors, perspectives and views correspond to stakeholder views, attribute relationships accommodate the idea of multilayered decomposition and metrics equals metrics. The quality meta-model does not, however, prescribe any visualization or attribute value calculation techniques. What is put forward in that respect by Georgiadou, is very useful. Her understanding of quality comes in a way close to the findings of this study. Quality factors equal with internal contributors in the meta-model and user needs are covered by the expected relationships between system and its context. The difference lies basically in that according to the meta-model, users or customers usually don't see the internal contributors and cannot have any precise expectations about them. Georgiadou (2003, 317) states that "quality factors are attributes that customers or users expect to find in the software". According to the meta-model quality is in the first place determined by the existence or degree of desired relationships between system and its context. These relationships can be more easily seen by different stakeholder types.

Wong and Jeffery (2002) discuss a model for software evaluation called "software quality evaluation framework" (SEF). It focuses on relationships between characteristics of software, desired consequences produced through usage and desired values achieved. The latter two influence the choice of characteristics used in software evaluation. Evaluators, in turn, are influenced by their job roles, the biggest difference being between users and developers. The values are for Wong and Jeffery things like 'job security' or 'self-fulfilment' and consequences like 'can do job faster' or 'easier to make decision'. Software characteristics used in the study were 'support', 'economic', 'institutional', 'usability', 'technical', 'functional' and 'operational'. The study shows by statistical analysis of survey results how the individual values, consequences and characteristics are related in users and developers minds respectively. Wong (2004) adds metrics to SEF. Measurement is based on questions like 'how satisfied are you with the reports' or 'ease in modifying the application to meet your business needs'. Wong (2004) sees a close relationship between SEF and goal question metric (GQM) method. Values correspond to goals, and the cognitive structures formed by relationships between characteristics, consequences and goals provide the questions.

Like GEQUAMO, SEF is more limited compared to the quality meta-model regarding aspects of quality modelling that are comprised by the framework. It does not, for example, discuss priorities, attribute relationships and use cases. SEF, however, puts in the clear focus the relationship between information system and its context in the form of desired consequences of usage and values that actors want to achieve. SEF also underlines the importance of user's perspective. Consequences can be compared to indicators in the meta-model and questions included into the instrument element of metrics sub-model. Wong and Jeffery (2002) and Wong (2004) note the variety of interpretations of quality, but don't make a choice or give their own alternative.

Côté et al. (2007) note the difficulties in defining the concept of quality, but don't give any alternative. The main goal of their paper is to set requirements for quality models and to evaluate four well known models according to them. No suggestions are, however, made about what should be the mandatory elements of quality models. Côté et al. (2007) set following three general requirements for quality models (ibid 2007, 405):

1) "A quality model should support the five different perspectives of quality as defined by Kitchenham and Pfleeger (1996)"
2) "A quality model … should allow for defining quality requirements and their further decomposition into appropriate quality characteristics, subcharacteristics and measures"
3) "A quality model … should allow for required measurements and subsequent aggregation and evaluation of obtained results"

According to Kitchenham and Pfleeger (1996) quality can be described from five perspectives: transcendental, user, manufacturing, product and value. This view was based on David Garvin's studies (Garvin 1984) on quality in different domains like philosophy, economics, etc. In transcendental view quality is an ideal that may never be achieved. The realized quality in each system implementation is just an approximation. From user-view perspective quality means meeting the user's needs in a task context. The manufacturing view looks at the development process and its compliance to standards, which should lead automatically to a quality product. Product view, in turn, means observing and measuring internal product properties. Finally, in value based view requirements and products are compared to their costs and potential benefits to the producer or user organization.

The proposed quality meta-model and system specific quality models created using it can accommodate four of the five perspectives listed above. The gap between ideal and practice can be handled with setting realistic goal values for quality attributes. Users' needs are embodied in the attribute definitions, especially in the definition of 'usefulness'. Involving different stakeholders into quality modelling allows for different needs and tastes. Product properties are taken into account as contributors to qualities. There are, however, other contributors too, among them even developers and their skills. And potential benefits and costs can be defined and discussed in connection with attributes like "affordability", "profitability", etc. The only perspective that has intentionally been left outside the quality meta-model is manufacturing view. It is regarded as being part of process quality model.

The second criterion for quality models set by Côté et al. (2007) requires that quality attributes can be decomposed into smaller subcharacteristics and measures. It is called "top to the bottom" approach. The proposed quality meta-model does not prescribe any "size" or level for the attribute or characteristic that is being modelled. One of the possible attribute relationships is the whole-part or aggregation relationship. In addition indicators as attribute definition elements decompose the attribute into smaller features. In addition the modelling process starts from general definition of domains (areas of concern) and attributes, which all can be made understandable to non-technical stakeholders. Next the process goes through use-cases and scenarios to verifiable detailed indicators (indicate to the users the existence of desired quality) and finally to technical and other contributors. In other words, there is in the model a clear and continuous, backwards traceable top-down path. It is important to keep in mind, however, that the meta-model does not assert any one-to-one relationship between internal contributors or development process features and observed and measured quality of information system.

The third and final criterion states that a proper quality model should allow for, i.e. include, measurements. The case studies did not focus on measuring quality, but showed that designing measurement arrangements suggests sometimes refinements to definition elements of the quality attribute in question. Consequently, some kind of metrics part is a useful attachment in product quality model. In the proposed meta-model it consists of following sub-elements: unit, scale, instrument, procedure and measured value. How subjective or objective the measurement arrangements and obtained values are, depends on the skills of the actors involved.

Jureta et al. (2009) focus on service-oriented systems (SOS) and create a quality model for them. This model can be called a meta-model because it is said to integrate relevant constructs for conceptualizing quality in SOS. In addition the authors refer explicitly to the ontology of quality. It is underlined that the proposed model does not itself define particular qualities and that it integrates two new sub-models, namely one for specifying priorities and one for specifying dependencies between qualities. The other sub-models are quality characteristics sub-model and quality value sub-model. All the four sub-models are given exact definitions using formal language.

The first sub-model in the theory presented by Jureta et al. (2009) is quality characteristics sub-model and it corresponds approximately to the quality attribute part of the quality meta-model. In addition to the concept of 'quality characteristic', it relies on the constructs 'quality dimension' and 'aggregation function'. A set of quality dimensions, like network time, selection time and execution time, can be aggregated by aggregation functions into a higher level dimension like latency. Then sets of quality dimensions can be mapped onto quality characteristics, in this case onto performance. Consequently, quality characteristic is defined as a set of distinct quality dimensions. Finally, sets of quality characteristics can be mapped onto one combined characteristic. A quality dimension is measurable and has a number of attributes: name, description, purpose, type, etc.

Quality dependency sub-model deals with dependencies between quality dimensions and quality priority sub-model with priority orders for pairs of quality dimensions or characteristics. The former shows how values of dimensions are related especially in order to help in managing tradeoffs, and the latter indicates which one of the dimensions (or characteristics) in pairs to optimize at the expense of the other. Finally, the quality value sub-model defines how quality is measured. It declares a set of possible values for a given quality dimension and states probabilities for that a certain value will be achieved provided that certain preconditions hold. In addition this sub-model describes post-conditions and preference orderings for different values.

Quality dimensions are comparable to indicators in the information system quality meta-model. Aggregation of indicators into quality attributes (characteristics in SOS quality model) is implicit in the meta-model. And grouping attributes according to some criteria is part of attribute relationships and domain elements. Jureta et al. (2009) analyse dependencies between quality dimensions in a separate sub-model, whereas the meta-model deals with quality attribute relationships on a higher level. Following the SOS quality model, introducing indicator relationships into meta-model could be useful. The quality priority sub-model, in turn, is comparable to the priority element of meta-model. Again, the latter does not cover priorities between indicators. Finally, the quality value sub-model coincides with metrics part of the meta-model, which does not, however, go into such details like probabilities and preconditions. All in all, the article by Jureta et al. (2009) demonstrates the benefits of sub-modelling, which has not been in the focus of this study. Another good feature is the formalism in construct definitions. What comes to comprehensiveness, the SOS quality model lacks certain sub-models and aspects of quality, like human actors, context and use cases.

Recently Frey et al. (2011) have put forward a quality meta-model to support design rationale and justify design decisions. The latter is defined as "explanation of why a designed artefact is the way it is" (Frey et al. 2011, 266). The mode is called QUIMERA (QUality metamodel to IMprove the dEsign

RAtionale). According to the authors quality can be seen from four perspectives: expected (client's needs), wished (what quality expert wants to achieve), achieved and perceived (client's perception of the end product) quality. Perspectives in turn are related to system lifecycle phases: specification (expected and wished), implementation (achieved) and use (perceived). The core of QUIMERA is composed of criteria (e.g. suitability for task, error tolerance, etc.) that are set for a certain artefact, and that can be decomposed into sub-criteria. Next, recommendations (e.g. "maximizing the number of criteria that are satisfied" Frey et al. 2011, 268) are specified for each criterion and given a weight to express their importance. The fulfilment of recommendations (and accordingly meeting the criteria) is evaluated through metrics or practices. The former offers numerical results and the latter logical (true or false) results. The meta-model can be used, for example, to support design decisions, i.e. selecting between design options by evaluating how well they conform to set criteria.

Comparing QUIMERA to the meta-model presented in the study at hand shows that criteria correspond to quality attributes, recommendations to goals, weight to priority, metrics and practices to metrics. With respect to the number elements QUIMERA is simple. The authors note, however that it is expandable. The four perspectives taken up by Frey et al. (2011) can be covered by implementing actor-perspective-view sub-model and relating it to other parts of the framework. Finally, the QUIMERA model does not provide any definition of quality. The good feature in the work of Frey et al. (2011) is the use of standard formal diagramming technique, namely UML, when presenting the meta-model. In the work at hand model diagrams are shown as they were drawn and used in the case studies. UML diagram drafts depicting the final meta-model can be found in Appendix 15 and 16. Raising the degree of formalism and sharpening the concepts is seen as part of further research.

The above review of newer initiatives for creating a meta-level or general understanding of information system quality models does not include everything that is written since ISO 9126 (2001). It treats, however the main ideas that have emerged. Compared to the quality meta-model proposed in the study at hand the alternatives are less comprehensive and do not provide a clear definition of quality. On the other hand some sub-models in the newer theories are more advanced. Further, the proposed meta-model leaves intentionally process quality and to some degree even metrics outside the focus.

8 CONCLUSION

This final chapter summarizes the results, implications and contributions of the study. It also takes up the limitations and finally lists the needs and possibilities for further research.

8.1 Main results and implications

An initial quality meta-model was successfully consolidated from traditional quality models and selected literature by generalization and conceptual analysis. Three case studies, where the meta-model was used as a template, proved in general its validity through usefulness of system specific models, their conformity to real world and comprehensibility to participants. The meta-model was also found to be comprehensive, general and flexible enough giving space to various instantiations of its elements in different contexts. At the same time the case studies and following analysis revealed some gaps and defects in the initial meta-model which resulted in element modifications, division into sub-models and reorganization of the model structure, and eventually in creation of the final model version.

A meta-model, that is needed to fully account for the quality of information system as a technical artefact in its context of development and use, is a hybrid model with six sub-models: 1) a model of actors, perspectives and views, appropriate models of 2) information system and 3) context, 4) a model of domain-attribute set, 5) a model a of quality attribute and 6) a metrics model. It can alternatively be described as an intersection of models. Domains are common or specific areas of concern with respect to operation, administration or further development of the target system. Views are views of system and context. The first and last sub-models are actually a part of development process model. Information system and context models are also models by themselves. The former comprises a relevant subset of architectural, design and other models of the system. All of them, however, contribute for their part to defining and understanding quality requirements. The pure quality sub-models, for designing overall quality and individual qualities of an information system as an end product of development process, are 1) a domain-attribute set and 2) individual attribute models. The three case studies focused especially on these two sub-models.

The context of an information system as a technical artefact can be divided into two major parts: 1) the immediate context of information processing in general, including human mind, cognitive processes, language and traditional manual information tools and 2) the variety of goal oriented human activities supported by the former. Most of the characteristics that are traditionally called 'quality attributes' refer to a particular desired relationship, or to a set of relationships, between the system and the two layered context. It can alternatively be called a 'desired state of affairs'. All internal system features get in the end there justification or "raison d'être" through these two system-context relationships. Consequently the *quality of an information system as a technical artefact in its context of development and use is in first place* determined by the existence and intensity of the desired relationships and therefore, must be defined and measured in terms of them. And the *design and realization of the desired state of affairs is the ultimate goal of quality modelling*. If even usefulness is seen as one of the core system qualities, all requirements can in fact be traced back to requirements about the relationship between information system and its context. The desired relationships between system and context can be described in detail using indicators in connection with use cases and scenarios, and on general level with entity-relationship models. The former are facts that in connection with real use cases and scenarios show the existence and intensity of the relationship in question. *In the second place the quality of an information system is* determined by so called contributors or factors which are things inside or outside the system that bring about and maintain the desired relationships.

Quality requirements constitute the core of requirements for an information system. The main difference between *quality requirements and requirements in general* is that the former have more importance to actors than the latter, and the latter can often be derived from the former. Therefore actors also set goals regarding to what degree these requirements need to be met. In principle, any attribute can be a quality attribute if it is of importance to the actors. Two other characteristics associated with quality requirements are that they usually pertain to relationships between system and context and their priorities and the measured values of respective attributes can vary according to different factors. Evidently this relativity radiates to the system in general. Further, all this emphasizes *the importance of actor sub-model*.

The overall quality of information system is determined by the implementation and realization of overall quality design embodied in the system specific domain-attribute set. It attributes the qualities to the system as a whole or to individual constituents, defines priorities and interrelationships between quality attributes and areas of concern, and sets balanced goal values. The system specific domain-attribute set can be regarded as the actual master plan or "DNA" of the information system. This plan undergoes usually many changes during system development process.

The case studies gave also clues to carrying out the quality modelling itself and embedding the quality modelling workflows and tasks into overall system development process. The results were presented in a tentative process model. If the overall quality design can be regarded as the master plan of the system, the *quality engineering process as a whole can be taken as the lead activity in a quality-driven system development*. This arrangement keeps the teams eyes focused on the desired relationships between the system and its context. Further, due to the relationship between quality requirements and requirements in general the proposed meta-model might be applicable to requirements capture and analysis in general.

The order in which elements of quality meta-model are applied is not arbitrary. The case studies suggest starting with selection of actors and awareness of their perspectives and views and then continuing with initial system and context descriptions. Next come selection and prioritization of domains, i.e. areas of concern with respect to the target system. Thereafter the easiest way to proceed for ordinary actors is identifying inside each area of concern positive and negative facts that would indicate existence or lack of quality respectively. On the basis of these indicators quality modelling can continue with the lead of an experienced editor and project manager. Facing numerous domains and attributes ordinary actors need a lot of guidance. *Use cases and scenarios can act as backbones of applying* remaining *meta-model elements*.

Comparison to theoretical background in Section 4.3 showed that the quality meta-model is positioned to a higher level of abstraction. In addition it comprises more important aspects of quality modelling than any of the background theories alone. Comparison of the final meta-model with the existing alternatives, in turn, showed, firstly, that the proposed meta-model complies with most general requirements for quality models set by Côté et al. (2007). Development process is, however, intentionally left outside of meta-model. The only exception is that actors (participants in the process), their perspectives and views are included as a useful sub-model. And in fact metrics (measurements) is also rather part of development process than product quality design. Secondly, the comparison showed that the proposed meta-model is more comprehensive than alternatives. On the other hand some of the models go more into details inside sub-model or inside elements. This may, however, easily turn into loss of generality and flexibility that are required from a meta-model.

8.2 Contributions

The main contribution of the study is that it combines the essential aspects of product quality in one comprehensive conceptual framework called the quality meta-model. In addition it points out that the essence of information system quality is in the existence and intensity of desired and actual relationships between a system and its context. The conceptual framework together with a refined understanding of quality generates a set of new questions to researchers. Developers, in turn, should direct more resources on understanding these relationships and let the realization of them lead the whole development process. This approach is called "quality driven development".

As a by-product the study outlines steps for applying the quality meta-model in concrete development projects. Finally the study gives practical examples of quality modelling in the infrastructural and cultural context of East-Africa. These experiences are of special interest to researchers and developers who need to work in similar environments.

8.3 Limitations

The study has been focusing more on how to design the quality of an information system than on how to implement and measure it. Further, the focus has been on the model of system quality as an end product of modelling process rather than on the process itself. The connections between system development process in general and the quality of resulting product have been excluded from the study.

Regarding levels of abstraction, the proposed quality model stays on a meta-level, i.e. on the level of abstract concepts. The degree of conceptualization is, however, not yet the highest possible. In addition the relationships between concepts are only logical. The study uses empirical observations about applying the conceptual framework to create actual system specific quality models for testing and evaluating the meta-model. It does not, however, assume anything about, for example, what is the correct set of quality attributes, the right diagramming technique, a valid way of carrying out measurements and calculating attribute values, etc. Dealing with these issues are studies in their own right.

A limitation of the research design itself is that only the proposed quality meta-model was tested. An option for research design would have been to conduct a parallel test of one or two alternative theoretical frameworks. A second limitation is that even with three case studies the observations are quite singular and from East-Africa only. In addition, the instantiated quality models were at the time of closing the cases still incomplete and more like prototypes. A perfect follow-up of model implementation and measurement of attribute values was not possible within the timeframes allocated for research work. Finally, objectivity, replicability and generalizability are the well known problems of case study method. The only solution is repeated application of the model in practice.

8.4 Future research

As a comprehensive model for product quality the meta-model and the case studies where it was applied open a wide range of questions for future research. Some of the most important are:

- How do contemporary system specific quality models take into account the essence of quality as desired system-context relationships?
- How to arrange quality attributes into categories? Can a widely acknowledged division into domains be achieved?
- Which actor groups can give the most relevant information concerning the quality of information system? Are some views and perspectives more informative and productive than others with respect to quality modelling?
- What are the most important axes in the frame of reference for modelling and measuring quality?
- How is the physical, infrastructural and cultural context reflected in quality models?

Based on the understanding of three levels of modelling – instance-, general- and meta-level – the circle from top down and back again can be iterated in order to reach better and better general level quality models for different system types and contexts. At the same time the degree of conceptualization will be raised. Of special interest, from the viewpoint of this study, are requirements for systems in environments where infrastructural setting is weaker than in highly industrialized countries. Do the imported systems reflect universal requirements or the requirements of their producers? But even more important is to get software companies interested in testing the meta-model and the idea of quality driven development. This could offer a genuine action research setting and an opportunity to refine quality modelling as a process. In connection with further research alternative quality modelling frameworks can also be used and compared to the one proposed in this study. Finally, application tools to assist in quality modelling could be created or tested.

YHTEENVETO (FINNISH SUMMARY)

Informaatioteknologialla on vuosikymmeniä pitkä historia teollistuneissa maissa. Tästä huolimatta käyttäjien tyytymättömyys päivittäisessä työssä käytettyihin järjestelmiin on varsin yleistä. Tämän takana on monia eri syitä. Ensinnäkin termi ”laatu” on muodikas, jopa ylikäytetty, ja usein ilman tarkkaa merkitystä. Toiseksi, perinteiset laatumallit mielletään helposti kahlitseviksi ja niiden käyttö suuritöiseksi. Kolmanneksi, normatiivisuudestaan huolimatta ne ovat olleet epätäydellisiä ja metriikkaan painottuneita. Laadullisten ongelmien yleisyys ja jatkuvuus vuosikymmenestä toiseen ei kuitenkaan ole vielä johtanut asiaa koskevaan tyhjentävään tieteelliseen keskusteluun ja lopullisiin teorioihin. Joitakin erittäin hyviä kirjoituksia on kuitenkin ilmestynyt. Käsillä olevan tutkielman tarkoituksena onkin tuoda vielä lisää näkökulmia keskusteluun ja esitellä entistä kokonaisvaltaisempi abstrakti viitekehys, nimeltään ”tietojärjestelmän metalaatumalli” (information system quality meta-model), yksityisen tietojärjestelmän laatumallin malli, tietoteknisen tuotteen tai järjestelmä laadun suunnitteluun ja mittaamiseen. Painopiste on kuitenkin laadun suunnittelussa.

Päätutkimuskysymyksenä on esille tuodun teoreettisen viitekehyksen validiteetti. Tutkimuksessa viitekehys luodaan aluksi perinteisten laatumallien ja suppean kirjallisuuskatsauksen pohjalta. Tämän jälkeen sitä testataan kolmessa erillisessä Itä-Afrikkaan sijoittuvassa tapaustutkimuksessa. Havaintojen perusteella mallista kehitetään lopullinen versio, jota verrataan uudempaan laatua ja laatumalleja koskevaan kirjallisuuteen. Yleinen lähestymistapa tutkimuksessa on yhdistelmä käsiteanalyysistä, teoriaa luovasta ja teoriaa arvioivasta tutkimuksesta. Tutkimus on myös pääosin kvalitatiivista tutkimusta. Koska teoreettinen tausta ei sisällä varsinaisia laatua koskevia meta-malleja, esitettyä viitekehystä on arvioitava muilla menetelmillä. Näitä ovat ensinnäkin sen tarjoamat suuntaviivat laadunsuunnittelu-toiminnalle ja ideat suunnittelua avustaviksi dokumenteiksi tai välineiksi. Toiseksi teoreettisen viitekehyksen sisältämien käsitteiden on vastattava todellisuutta. Edelleen, teorian on oltava ymmärrettävä laatumallin rakentamiseen osallistuville, riittävän yleinen salliakseen vaihtoehtoja mallin soveltamisessa ja ja lisäksi oltava avuksi laadun suunnittelutyön hallinnassa.

Kaikki kolme erillistä ja erilaisia tietojärjestelmiä koskevaa tapaustutkimusta tukevat yllä mainittujen kriteerien näkökulmasta esitetyn teoreettisen viitekehyksen yleistä validiutta. Meta-malli on hybridi malli, joka koostuu kuudesta osamallista: 1) toimijat, heidän perspektiivinsä ja näkymänsä, 2) tietojärjestelmä, 3) konteksti, 4) ryhmitelty ja priorisoitu atribuuttijoukko, 5) yksityisen atribuutin malli ja 6) malli yksityisten attribuuttien arvojen ja kokonaislaadun mittausta varten. Keskeisimmät mallit laadun suunnittelun kannalta ovat mallit 4 ja 5. Vertailu viimeaikaisessa kirjallisuudessa esitettyihin vaihtoehtoihin osoittaa muodostetun viitekehyksen olevan näitä laajempi. Toisaalta se ei nykyisessä muodossaan yllä osamallien yksityiskohdissa ja formalismissa samalle tasolle kuin eräät suppeammat mallit.

Perinteisten laatuatribuuttien analyysi osoittaa, että suurin osa niistä itse asiassa viittaa johonkin tai joihinkin toivottuihin suhteisiin tietojärjestelmän ja kontekstin välillä. Tämän johdosta myös tietojärjestelmän laatu tulisi tutkimuksen mukaan ensisijaisesti määritellä ja mitata näiden suhteiden kautta, ja näiden suhteiden suunnittelu ja realisointi tulisi vastaavasti olla laadunhallinan varsinainen päämäärä. Vasta toissijaisesti, mutta silti väistämättömästi, tietojärjestelmän laatua tulee tarkastella niin sanottujen vaikuttavien tekijöiden kautta, joita löytyy niin tietojärjestelmän ulko- kuin sisäpuoleltakin. Nämä tekijät auttavat realisoimaan suunnitellun laadun. Kaiken kaikkiaan laatuvaatimukset muodostavat tietojärjestelmää koskevien vaatimusten ytimen. Ne voivat olla suhtellisia ja toimijakohtaisia, mutta ne ovat niin tärkeitä, että ne yleensä priorisoidaan ja niille asetetaan tavoitearvot. Käsillä olevassa tutkimuksessa priorisoitua atribuuttijoukkoa kutsutaan tietojärjestelmän DNA:ksi ja sitä ”johtotähtenään” pitävää järjestelmäkehitystä laatuvetoiseksi kehittämiseksi.

Edellä esitettyyn perustuen tutkimuksen päätuloksia ovat tietojärjestelmän laatusuunnittelun teoreettinen viitekehys eli meta-malli ja laadun käsitteen analyysi tietojärjestelmän ja kontekstin välisen suhteen näkökulmasta. Lisäksi tutkimus kuvaa laadunmäärittelyn vaihejakoa ja tarjoaa tapauskuvauksia laadunmäärittelystä Itä-Afrikkalaisessa ympäristössä. Tutkimus ja sen tulokset avaavat samalla runsaan joukon uusia tutkimuskysymyksiä. Itse meta-malli on myös avoin jatkokehitykselle.

REFERENCES

Ahmed K. Z., Umrysh C. E.. (2002). *Developing Enterprise Java Applications with J2EE and UML.* Boston: Addison-Wesley.

Ali R., Dalpiaz F., Giorgini P. (2010). A goal-based framework for contextual requirements modeling and analysis. *Requirements Engineering. 15(4),* 439-458.

Allen D., Karanasios S., Slavova M. (2011). Working With Activity Theory: Context, Technology and Information Behavior. *Journal of the American Society for Information Science and Technology. 62(4),* 776-788.

Alter S. (1999). A General, Yet Useful Theory of Information Systems. *Communications of the Association for Information Systems. 1(3),* Art. 13.

Alter S. (2008). Defining information systems as work systems: implications for the IS field. *European Journal of Information Systems, 17(5),* 448-469.

Anon (2010). The Information Society and ICT Sector Development Project – TANZICT. Ministry of Foreign Affairs of Finland, Tanzanian Ministry of Communications, Science and Technology. Project document. Draft 16.5.2010.

Anon (2011). Tanzania power cuts worsen after drought. [15.02.2011] Thomson Reuters. Retrieved from: http://af.reuters.com/article/investingNews/idAFJOE71E0HR20110215

Bagozzi R. (2007). The Legacy of the Technology Acceptance Model and a Proposal for a Paradigm Shift. *Journal of the Association of Information Systems, 8 (4)*, 244-254.

Basili V. (1993). Applying the Goal/Question/Metric Paradigm in the Experience Factory. *Proceedings of the Tenth Annual CSR (Centre for Software Reliability) Workshop, Holland, September 29 - October 1, 1993*.

Basili V., Caldiera G., Rombach H.D. (1994), Goal Question Metric Approach, in J.J. Marciniak (ed.) *Encyclopedia of Software Engineering*. New York: John Wiley & Sons, Inc. 528-532.

Baskerville R. (1999). Investigating information systems with action research. *Communications of the Association for Information Systems. 2(3),* Art. 4.

Bass L., Clements P., Kazman R. (2003). *Software Architecture in Practice.* Boston: Pearson Education.

Behkamal B., Kahani M., Akbari M. (2009). Customizing ISO 9126 quality model for avaluation of B2B applications. *Information and Software Technology 51 (3)*, 599-609.

Benbasat I., Goldstein D., Mead M. (1987). The Case Research Strategy in Studies of Information Systems. *MIS Quarterly, 11(3),* 369-386.

Berki E., Georgiadou E., Holcombe M. (2004). Requirements Engineering and Process Modelling in Software Quality Management – Towards a Generic Process Metamodel. *Software Quality Journal, 12,* 265-283.

Boehm, B.W., Brown, J.R., Kaspar, H., Lipow M., McCleod, G.J., and Merritt M.J. (1978). *Characteristics of Software Quality*. Amsterdam: North-Holland.

Bollinger T., Voas J., Boasson M. (2004). Persistent Software Attributes. *IEEE Software*, *21(6)*, 16-18.

Buschmann F. (2011). Unusable Software Is Useless, Part1. *IEEE Software, 28 (1)*, 92-94.

Callaos N., Callaos B. (1996). Designing with a systemic total quality. In N. Callaos (ed.) *Proceedings of the International Conference on Information Systems Analysis and Synthesis, ISAS'96,* Orlando, Fla: International Institute of Informatics and Systemics.

Cambridge Advanced Learner's Dictionary (2008). Third Edition. Cambridge: Cambridge University Press.

Côté M-A., Suryn W., Georgiadou E. (2007). In search for a widely applicable and accepted software quality model for software quality engineering. *Software Quality Journal, 15(4),* 401-416.

Chua B. B., Dyson L. E. (2004). Applying the ISO 9126 Model to the Evaluation of an e-Learning system. In R. Atkinson, C. McBeath, D.Jonas-Dwyer, R. Phillips (eds.), *Beyond the Comfort Zone: Proceedings of the 21st ASCILITE Conference*. Perth, 5-8 December, 184-190.

Chung L., Nixon B., Yu E., Mylopoulos J. (1999), *Non-FunctionalRequirements in Software Engineering*. Springer.

Crispin L. (2006). Driving Software Quality: How Test-Driven Development Impacts Software Quality. *IEEE Software, 23(6),* 70-71.

Cysneiros L. M., Leite J.C.S.P. (2004). Nonfunctional Requirements: From Elicitation to Conceptual Models. *IEEE Transactions on Software Engineering, 30 (5)*, 328-350.

Dahlberg T., Järvinen J. (1997). Challenges to IS quality. *Information and Software Technology, 39(12),* 809-818.

de Bruin H., van Vliet H. (2003). Quality Driven Software Architecture Composition. *The Journal of Systems and Software, 66(3)*, 269-284.

Denger C., Shull F. (2007). A Practical Approach for Quality-Driven Inspections. *IEEE Software, 24(2),* 79-86.

Dromey, G. (1996). Cornering the chimera, *IEEE Software , 13(1)*, 33–43.

Dubin R. (1969). *Theory building.* NewYork: The Free Press.

Egyed A., Grünbacher P. (2004). Identifying Requirements Conflict and Cooperation: How Quality Attributes and Automated Traceability Can Help. *IEEE Software, 21(6),* 50-58.

Eriksson I., Törn A. (1991). A model for IS quality. *IEEE Software Engineering Journal, 6(4),* 152-158.

Fenton N.E., Pfleeger S.L. (1998). *Software Metrics: A Rigorous and Practical Approach.* PWS Publishing.

Finne A. (2005). Blueprint for General Quality Attribute Model. In E. Hustad, B. Munkvold, K. Rolland, L. Flak (eds.) *Proceedings of the 28th Information Systems Research Seminar in Scandinavia, Kristiansand, Norway, August 6-9.*

Finne A. (2006). Applying a Twofold Quality Model: Producing Groundwork for System Specific Attribute Models. In H. Hautop, E. Sutinen, M. Duveskog, Kinshuk, A. Mkocha. (eds.) *Proceedings of the Fourth IEEE*

International Workshop on Technology for Education in Developing Countries, Iringa, Tanzania, July 10-12. Los Alamitos, CA: IEEE Computer Society, 3-4.

Finne A. (2011). Towards a Quality Meta-Model for Information Systems. *Software Quality Journal, 19(4),* 663-688.

Frey A.G., Céret E., Dupuy-Chessa S., Cavalry G. (2011). QUIMERA: A Quality Metamodel to Improve Design Rationale. *Proceedings of the third ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS 2011),* ACM Press, Pisa.

Folmer E., Bosch J. (2007). A pattern framework for software quality assessment and tradeoff analysis. *International Journal of Software Engineering and Knowledge Engineering, 17(4),* 515-538.

García F., Bertoa M., Calero C., Vallecillo A., Ruíz F., Piattini M., Genero M. (2006). Towards a consistent terminology for software measurement. *Information and Software Technology, 48(8)*, 631-644.

Garvin D. (1984). What Does "Product Quality" Really Mean? *Sloan Management Review, 26(1),* 25-45.

Georgiadou E. (2003). GEQUAMO – A Generic, Multilayered, Customisable, Software Quality Model. *Software Quality Journal, 1,* 313-3123.

Glinz M. (2008). A Risk-Based, Value-Oriented Approach to Quality Requirements. *IEEE Software, 25(2)*, 34-41.

Grady R., Caswell D. (1987). *Software Metrics: Establishing a Company-wide Program.* New Jersey: Prentice Hall.

Gregor S. (2006). The nature of Theory in Information systems. *MIS Quarterly, 30(3)*, 611-642.

Hevner A., March S., Park J., Ram S. (2004). Design Science in Information Systems Research. *MIS Quarterly, 28(1)*, 75-105.

Holmström J., Sawyer S. (2011). Requirements engineering blinders: exploring information systems developers' black-boxing of the emergent character of requirements. *European Journal of Information Systems, 20(1)*, 34-47.

Holzer, M., Kim S.-T. (2007), *Digital Governance in Municipalities Worldwide (2007): A Longitudinal Assessment of Municipal Websites Throughout the World*. Newark: The E-Governance Institute, Rutgers University.

Huang L., Boehm B. (2006). How Much Software Quality Investment Is Enough: A Value-Based Approach. *IEEE Software, 23(5),* 88-95.

IEEE Standard Glossary of Software Engineering Terminology (1990). New York: IEEE.

Ingram D. (2009). *Design-Build-Run: Applied Practices and Principles for Production-Ready Software Development*. Indianapolis: Wiley Publishing, Inc.

ISO 8402 (1994). *Quality management and quality assurance – Vocabulary*. Geneva: ISO/IEC.

ISO 9000 (2000). *Quality management systems – Fundamentals and vocabulary*. Geneva: ISO/IEC.

ISO 9001 (2000). *Software engineering - Guidelines for the application of ISO 9001:2000 to computer software*. Geneva: ISO/IEC.

ISO 9126-1 (2001). *Software Engineering –Product Quality – Part 1: Quality Model*. Geneva: ISO/IEC.

ISO 15504-2 (1998). *Information technology –Process assessment – Part 2: A reference model for processes and process capability*. Geneva: ISO/IEC.
ISO 90003 (2004). *Software engineering - Guidelines for the application of ISO 9001:2000 to computer software*. Geneva: ISO/IEC.
ISO 12207 (2008). *Systems and software engineering – Software life cycle processes*. Geneva: ISO/IEC.
Jacobson I., Booch G., Rumbaugh J. (1999). *The Unified Software Development Process.* Boston: Addison-Wesley.
Jarke M., Loucopoulos P., Lyytinen K., Mylopoulos J. (2011). The brave new world of design requirements. *Information Systems, 36(7),* 992-1008.
Jung H-W., Kim S-G., Chung C-S. (2004). Measuring Software Product Quality: A Survey of ISO/IEC 9126. *IEEE Software, 21(5),* 88-92.
Jureta I.J., Herssens C., Faulkner S. (2009). A comprehensive quality model for service-oriented systems. *Software Quality Journal, 17(1),* 65-98.
Järvinen P. (2001). *On Research Methods.* Tampere: Opinpajan kirja.
Kaplan A. (1964). *The Conduct of Inquiry: Methodology for Behavioral Science.* San Francisco: Chandler Publ.
Karlsson J., Ryan K.. (1997). A Cost-Value Approach for Prioritizing Requirements. *IEEE Software,14(5)*, 67-74.
Kazman R., Bass L., Klein M., Lattanze T., Northrop L. (2005). A Basis for Analyzing Software Architecture Analysis Methods. *Software Quality Journal,13(4)*, 329-355.
Kearns L., Falls M. (eds.) (2004). *Inside the Minds: The Software Business*. USA: Aspatore, Inc.
Kelegai L., Middleton M. (2004). Factors influencing information systems success in Papua New Guinea organizations: A case analysis. *Australasian Journal of Information Systems, 11(2)*, 57-69.
Khaddaj S., Horgan G. (2005). A Proposed Adaptable Quality Model for Software Quality Assurance. *Journal of Computer Sciences, 1(4)*, 482-487.
Kinreich I. (2004). Software – The Power of the Invisible. In L. Kearns, M. Falls (eds.). *Inside the Minds: The Software Business*. USA: Aspatore, Inc. 111-119.
Kitchenham P., Pfleeger S.L. (1996). Software Quality: The Elusive Target. *IEEE Software,13(1)*, 12-21.
Kotonya G., Sommerville I. (1996). Requirement engineering with viewpoints. *IEEE Software Engineering Journal, 11(1),* 5-18.
Krishna S., Walsham G. (2005). Implementing Public Information Systems in Developing Countries: Learning From a Success Story. *Information Technology for Development, 11(2),* 123-140.
Kujala S., Väänänen-Vainio-Mattila K. (2008). Value of Information Systems and Products: Understanding the User's Perspective and Values. *Journal of Information Technology Theory and Application , 9(4),* 23-39.
Lagsten J. (2011). Evaluating Information Systems according to Stakeholders: A Pragmatic Perspective and Method. *Electronic Journal of Information Systems Evaluation,14(1)*, 73-88.

Lee A. (1989). A Scientific Methodology for MIS Case Studies. *MIS Quarterly, 13(1),* 33-50.

López C., Inostroza P., Cysneiros L., Astudillo H. (2009). Visualization and comparison of architecture rationale with semantic web technologies. *The Journal of Systems and Software, 82(8),* 1198-1210.

Maciaszek L.A., Liong B.L. (2005). *Practical Software Engineering:A Case Study Approach*. Harlow: Pearson Education Limited.

Macome E. (2008). On Implementation of an Information System in the Mozambican Context: The EDM Case Viewed Through ANT Lenses. *Information Technology for Development, 14(2),* 154-170.

Markus M.L. (1983). Power, Politics and MIS Implementation. *Comm ACM, 26 (6),* 430-444.

Martin J., Odell J.J. (1996). *Object-Oriented Methods: Pragmatic Considerations.* New-Jersey: Prentice Hall P T R.

McCall, J.A., Richards, P.K., and Walters, G.F. 1977. *Factors in Software Quality*, Vols. I–III, AD/A-049-014/015/055. Springfield, VA: National Technical Information Service.

Miyoshi T., Azuma M. (1993). An Empirical Study of Evaluating Software Development Environment Quality. *IEEE Transactions on Software Engineering, 19(5),* 425-435.

Muhren W., van den Eede G., van de Walle B. (2008). Sensemaking and Implications for Information Systems Design: Findings From the Democratic Republic of Congo's Ongoing Crisis. *Information Technology for Development, 14(3),* 197-212.

Narasimhaiah G., Lin S-L. (2010). Determinants of software quality: A survey of information systems project managers. *Information and Software Technology 52 (6)*, 602-610.

Nelson R., Todd P., Wixom B. (2005). Antecedents of Information and System Quality: An Empirical Examination Within the Context of Data Warehousing. *Journal of Management Information Systems, 21 (4)*, 199-235.

Nielsen J. (1993). *Usability Engineering*. Boston: Academic Press.

Niemelä E., Immonen A. (2007). Capturing quality requirements of product family architecture. *Software Information and Software Technology, 49(11-12),* 1107-1120.

Ortega M., Pérez M., Rojas T. (2003). Construction of a Systemic Quality Model for Evaluating a Software Product. *Software Quality Journal, 11(3),* 219-242.

Parviainen P., Tihinen M. (2007). A survey of existing requirements engineering technologies and their coverage. *International Journal of Software Engineering and Knowledge Engineering, 17(6),* 827-850.

Reeves C., Bednar D. (1994). Defining quality: Alternatives and implications. *The Academy of Management Review, 19(3),* 419-445.

Rincon G., Alvarez M., Perez M., Hernandez S. (2005). A discrete-event simulation and continuous software evaluation on a systemic quality model: An oil industry case. *Information & Management, 42(8),* 1051-1066.

Saaty T.L. (1980). *The Analytic Hierarchy Process*. New York: McGraw-Hill.

Savolainen R. (2006). Information use as gap-bridging: The viewpoint of sense-making methodology. *Journal of the American Society for Information Science and Technology, 57(8),* 1116-1125.

Scholl H., Eisenberg M., Dirks L., Carlson T. (2011). The TEDS Framework for Assessing Information Systems From a Human Actor's Perspective: Extending and Repurposing Taylor's Value-Added Model. *Journal of the American Society for Information Science and Technology, 64(4),* 789-804.

Seffah A., Kececi N., Donoyaee M. (2001). QUIM: A Framework for Quantifying Usability Metrics in Software Quality Models. In D. Young (ed.) *Proceedings of 2nd Asia-Pacific Conference on Quality Software (APAQS 2001).* Los Alamitos, CA: IEEE Computer Society, 311-318.

Shull F., Seaman C., Zelkowitz M. (2006). Victor R. Basili's Contributions to Software Quality. *IEEE Software, 23(1),* 16-18.

Siakas, K.V., Berki E., Georgiadou E. & Sadler C. (1997). The complete alphabet of quality software systems: Conflicts and compromises. *Proceedings of 7th World Congress on Total Quality & Qualex 97.* New Delhi: Tata McGraw-Hill Publishing Company, 603-618.

Siakas K.V., Georgiadou E. (2005). *PERFUMES. A Scent of Product Quality Characteristics.* SQM, UK.

Silva L. (2007). Institutionalization Does Not Occur By Degree: Institutional Obstacles in Implementing a Land Administration System in a Developing Country. *Information Technology for Development, 13 (1),* 27-48.

Simon H.A. (1996). *The Sciences of the Artificial*. Massachusetts: MIT Press.

Srinivasan R. (2007). Ethnomethodological Architectures: Information Systems Driven by Cultural and Community Visions. *Journal of the American Society for Information Science and Technology, 58(5),* 723-733.

Svahnberg M., Wohlin C., Lundberg L., Mattsson M. (2003). A Quality-Driven Decision-Support Method for Identifying Software Architecture Candidates. *International Journal of Software Engineering and Knowledge Engineering, 13(5),* 547-573.

Thanasankit T., Corbitt B. (2000). Cultural Context and its Impact on Requirements Elicitation in Thailand. *The Electronic Journal on Information Systems in Developing Countries, 1(2),* 1-19.

Tian J. (2004). Quality Evaluation Models and Measurements. *IEEE Software, 21(3),* 84-91.

Urquhart C., Lehmann H., Myers M. (2010). Putting the 'theory' back into grounded theory: guidelines for grounded theory studies in information systems. *Information Systems Journal, 20(4),* 357-381.

van Reijswoud V. (2009). Appropriate ICT as a Tool to Increase Effectiveness in ICT4D: Theoretical Considerations and Illustrating Cases. *The Electronic Journal on Information Systems in Developing Countries, 38(9),* 1-18.

Vesisenaho M. (2007). *Developing University-level Introductory ICT Education in Tanzania: A Contextualized Approach.* Doctoral Dissertation. University of Joensuu: Computer Science, Dissertations 16.

Villalba M., Fernández-Sanz L., Cuadrado-Gallego J., Martínez J. (2010). Software quality evaluation for security COTS products. *International Journal of Software Engineering and Knowledge Engineering, 20(1),* 27-48.

Voas J. (2004). Software's Secret Sauce: The "-ilities". *IEEE Software, 21(6),* 14-15.

Voas J., Agresti W.W. (2004). Software Quality from a Behavioral Perspective. *IT Professional, 6(4),* 46-50.

Wagner S. (2010). A bayesian network approach to assess and predict software quality using activity-based quality models. *Information and Software Technology, 52 (11),* 1230-1241.

Wand Y., Wang R.Y. (1996). Anchoring Data Quality Dimensions in Ontological Foundations. *Communications of the ACM, 39(11),* 86-95.

Weick K.E. (1984). Theoretical Assumptions and Research Methodology Selection. In F.W. McFarlan (Ed.) *The Information Systems Research Challenge*. Boston: Harvard Business School Press, 111-132.

Wixom B.H., Todd P.A. (2005). A Theoretical Integration of User Satisfaction and Technology Acceptance. *Information Systems Research, 16(1),* 85-102.

Wong B. (2004). The software evaluation framework 'SEF' extended. *Information and Software Technology, 46(15)*, 1037-1047.

Wong B., Jeffery R. (2002). A Framework for Software Quality Evaluation. In M. Oivo, S. Komi-Sirviö S. (Eds.) *Proceedings of PROFES 2002,* Lecture Notes in computer Science 2559, Springer, 103-118.

WTDR (2010) - (World Telecommunication/ICT Development Report): MONITORING THE WSIS TARGETS: A mid-term review. Geneva: ITU.

Yin R. (2009). *Case Study Research: Design and Methods.* Los Angeles: SAGE Publications.

Yonazi J. (2009). ICT4D: Facing the Challenges Head-on in Tanzania. [25.05.2011] The Institute of Finance Management. Dar es Salaam, Tanzania. Retrieved from: http://www.ngopulse.org/files/tanzania_thetha_report_ict4d.pdf

Özkan S. (2006). A Process Capability Approach to Information Systems Effectiveness Evaluation. *Electronic Journal of Information Systems Evaluation, 9(1)*, 7-14.

APPENDIX 1: RESEARCH PERMIT IN TANZANIA

TANZANIA COMMISSION FOR SCIENCE AND TECHNOLOGY
(COSTECH)

Telegrams: COSTECH
Telephones: (255 - 22) 2700745-6
Director General: (255 - 22) 2700750
Fax: (255 - 22) 2775313
Telex: 41177 UTAFITI
E-M: Rclearance@costech.or.tz

Ali Hassan Mwinyi Road
Kijitonyama Area
P.O. Box 4302
Dar es Salaam
Tanzania

RESEARCH PERMIT

No. 2004 – 58– NA– 2004 – 04 **Date:** 3rd **May 2004**

1. Name : **Auvo A. Finne**

2 Nationality : **Finnish**

3. Title : **Towards more Sustaining Information Systems. A Refined View on Requirements, Analysis, Design and Implementation of Implementation of Information Systems**

4. Research shall be confined to the following region(s) **Arusha, Manyara, Dar es Salaam, Dodoma and Iringa**

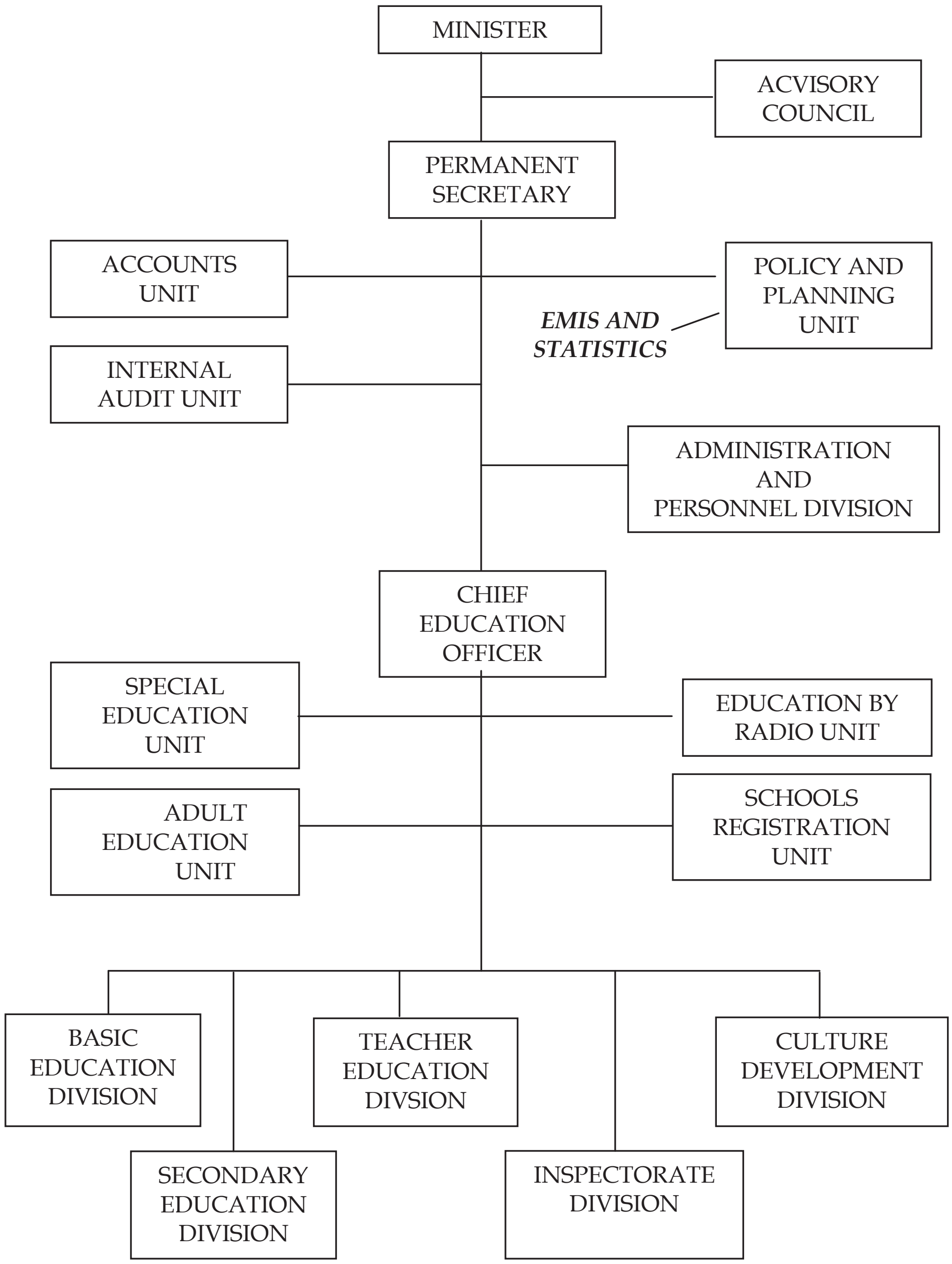
5. Permit validity: **3rd May to 2nd May 2005**

6. Local Contact/Collaborator: **Prof. B. Mutagahywa, Computing Centre, University of Dar es Salaam**

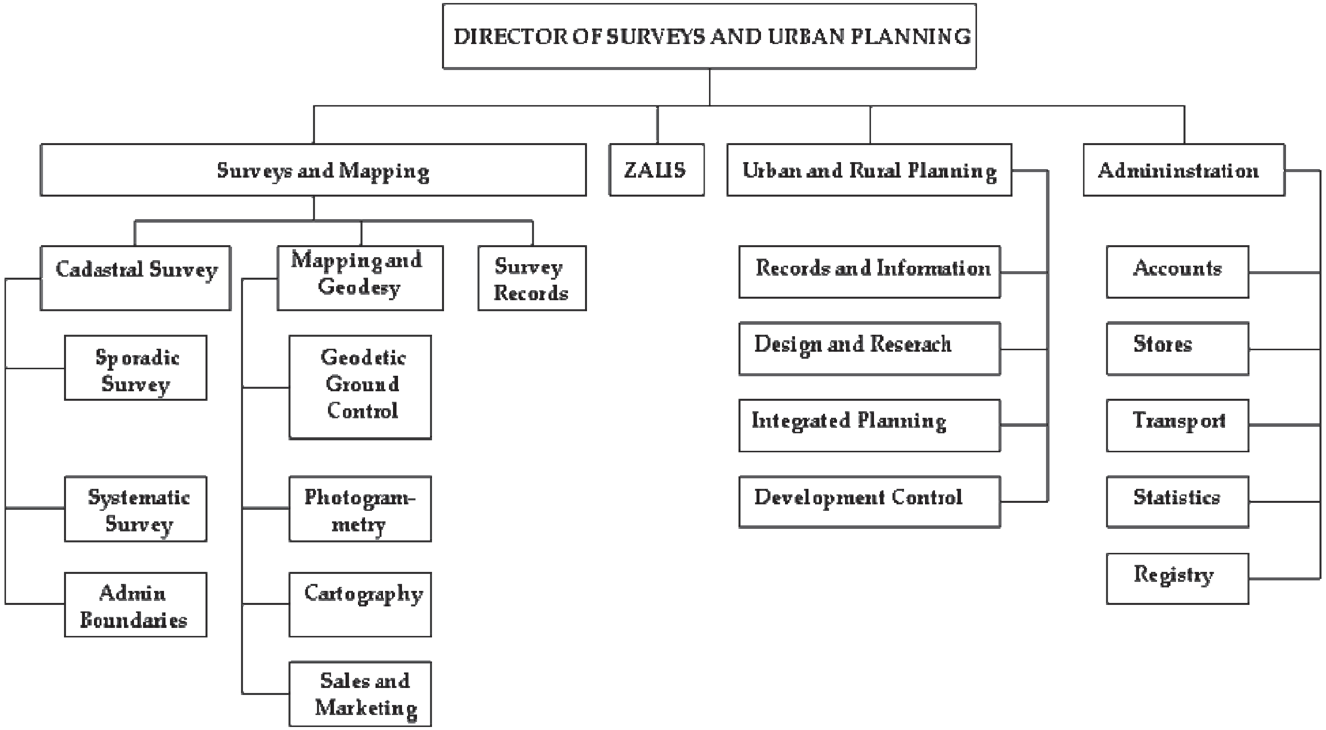
7. Researcher is required to submit progress report on quarterly basis and submit all Publications made after research.

H.P. Gideon
for: DIRECTOR GENERAL

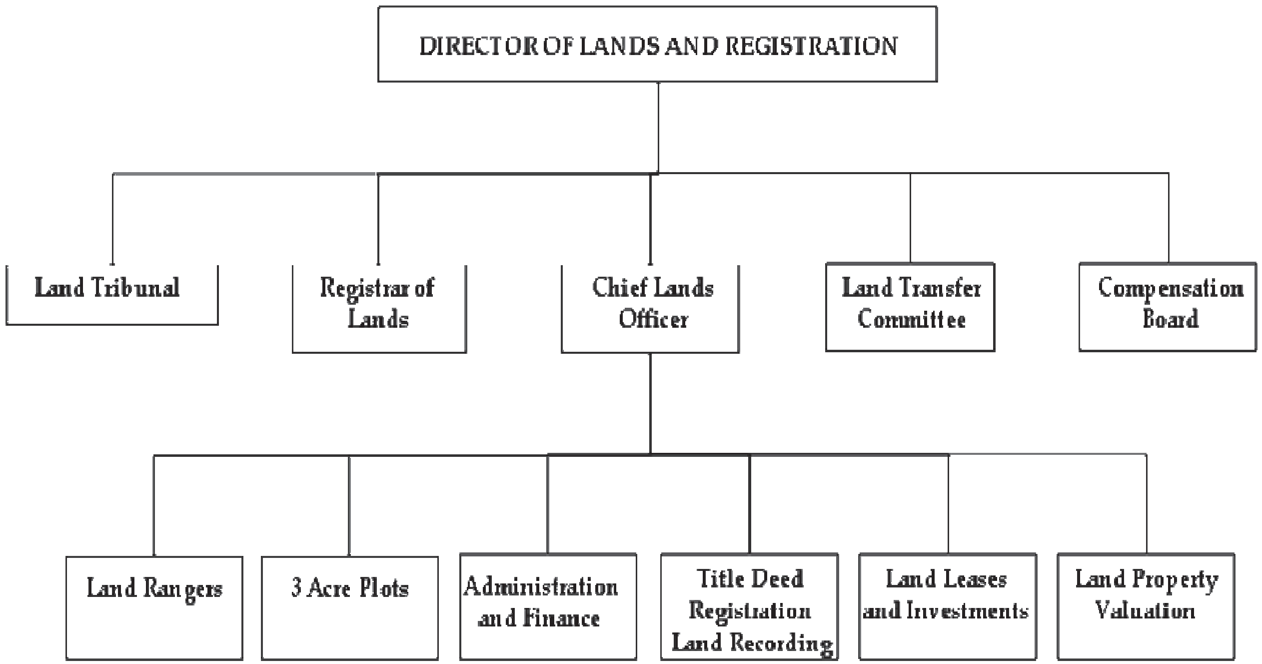
APPENDIX 2: ORGANIZATION STRUCTURE OF MINISTRY OF EDUCATION AND CULTURE IN TANZANIA



APPENDIX 3: ORGANIZATION STRUCTURE OF DEPARTMENT OF SURVEYS AND URBAN PLANNING IN ZANZIBAR



APPENDIX 4: ORGANIZATION STRUCTURE OF DEPARTMENT OF LANDS AND REGISTRATION IN ZANZIBAR



APPENDIX 5: GLOSSARY OF DOMAINS AND QUALITY ATTRIBUTES USED IN EMIS CASE

DOMAIN	ATTRIBUTE	DEFINITION
support to business processes of user organization	Each "business" process (PB) is a collection of activities for producing something of value to MoEC and its "customers". For example, in department of primary education all activities aimed at issuing guidelines on the quality of instructional materials constitute one business process. An information system, like EMIS, must assist in conducting the activities of particular business processes.	
	being needed	if EMIS is needed, you cannot successfully achieve what you want without it
	comprehensiveness	EMIS includes everything that is needed or relevant to PBs
	appropriateness	EMIS is suitable or acceptable for a particular situation and PBs
	timeliness	EMIS operates at a moment when it is useful, effective or relevant to PBs
	coordination	EMIS and business processes work together efficiently
	type and degree of use	how and how much EMIS is actually being used
	usefulness	you can use EMIS to do something; EMIS can help you to achieve something in a particular situation
	predictability	it is obvious in advance that EMIS will operate in a particular way at a certain time with respect to PBs
	regularity	EMIS operates always the same way and at the same time with respect to PBs
HC interaction (HCI (human-computer interaction)	It is about the presentation of information on the screen and the means of controlling and operating the computer. It's about how the computer	
	clearness	EMIS is easy to undestand; its features and functions are impossible to be mistaken about
	ease of use	you use EMIS without difficulty or effort; it is not complicated and causes no problems
	user orientation	concern with and for users has been in the minds of the designers of EMIS
	user friendliness	EMIS is well designed and easy to use
	accessibility	EMIS is easy to get into it, obtain it and use it
	simplicity	EMIS is not complicated; it has not unnecessary parts and details
	availability	you can obtain and find EMIS; its free for use
authorization	The authorization system protects computer resources by grating users selective authority to use individual files, data items, programs and devices.	
deployment	Deploying an information system means putting it in place so that it is ready to be used. The term refers mainly to delivering and istalling hardware and software.	
	installability	EMIS is easy to instal

(continues)

GLOSSARY OF DOMAINS AND QUALITY ATTRIBUTES USED IN EMIS CASE (continues)

DOMAIN	ATTRIBUTE	DEFINITION
architecture and design of IS	Architecture is concerned with system's components or subsystems and relationships among these element. It is a general level sketch of the system. It describes the coarse grain components of the system and covers both hardware and software. Detailed design, in turn, takes into consideration the platform on which the system is implemented. This means programming languages, operating systems, database management systems, etc.	
maintenance	Maintenance means keeping information system running and in good condition. It includes regularly checkint it and repairing when necessary.	
change in IS	Every information system experiences unitended or intrinsic change (growth of database in size, deterioration, etc.). Sometimes it instigates maintenance operations. Secondly, change means intentional product evolution.	
change in environment	The environment of an information system can be divided into at least three layers: user, user organization, society. All of these layers change constantly and cause pressure on developers to change the system itself.	
change of environment	(Sometimes an information system has to be moved to a different environment. In computer science this is called porting (adaptation of a piece of software so that it will function in a different computing environment to that for which it was originally written).	
impact on environment	An information system may have impact on its user, user organization or the society as a whole. Influence can be positive or negative.	
	type and degree of impact on environment	EMIS has effect on environment
data, information, contents	This domain embraces simply all kinds of contents (database records, text, pictures, video, documents, etc.).	
	coverage	the set of different information or categories of information contained in EMIS
	reliability	information in EMIS is likely to be correct
	validity	data in EMIS is what it is supposed to be; it is worth collecting and describes accurately reality
	accuracy	data in EMIS is correct, even in small details
	adequacy	data in EMIS is good enough or great enough in amount to be acceptable or used
	correctness	data in EMIS is in accordance with facts and has no mistakes
	relevancy	data in EMIS bears upon or is connected with the matter at hand; important, significant or approriate in a situation
	non-overlap	data in EMIS is not involving same information several times
	being up-to-date	EMIS has the latest information about things

(continues)

GLOSSARY OF DOMAINS AND QUALITY ATTRIBUTES USED IN EMIS CASE (continues)

DOMAIN	ATTRIBUTE	DEFINITION
regulations, standards, conventions, laws	Some environmental factors affect the system even though the system does not rely on them directly. Intangible aspects of the context include explicit rules and requirements such as government regulations, industry standards, and organizational policies.	
	compliance	EMIS is designed and acting according to what is required by regulations, standards, etc.
risk, harm to IS	Different things can cause physical harm to the information system. They include human actions as well as other factors like electricity surges. The harm may appear in the form of loss of data, lost network connection etc.	
	security (safety)	EMIS is safe from harm or danger
risk, harm to environment	This is specific and negative kind of impact on environment. The harm may meet users, user organizations, society, or the physical environment.	
	environment friendliness	the environment is safe from harm or danger caused by EMIS
failure, fault, error in functionality	The information system does not succeed in doing or achieving something.	
	reliability	EMIS can be trusted to work well and behave in the (usually repeatedly the same) way you want to
use of time and resources, performance	This domain could also be called efficiency, the ability to perform tasks without wasting time or other resources.)	
interaction with and integration to other ISs	Modern information systems usually act together with other information systems. This poses certain requirements to the parts of interaction.	
	compatibility	EMIS works well together with other systems; can exist together successfully; can be used together
	sharing	two or more people, programs, ISs... can use EMIS together
composite attributes	Presuppose the existence of a number of other attributes. E.g. to be sustainable, a system must also be affordable, installable, supported, etc.	
	sustainability	EMIS can continue functioning at the same rate or level without any problems
	suitability	EMIS is right and acceptable for a particular purpose

APPENDIX 6: GLOSSARY OF DOMAINS USED IN WEBSITE CASE

DOMAIN	DEFINITION
support to business processes of user organization	Each "business" process (PB) is a collection of activities for producing something of value to LWFMZ and its "customers". For example, in department of primary education all activities aimed at issuing guidelines on the quality of instructional materials constitute one business process. An information system, like website, must assist in conducting the activities of particular business processes.
entertainment	Information systems can, besides their usefulness, give users pleasure, amuse them, interest them, etc. Well known examples are computer games.
authorization	The authorization system protects computer resources by grating users selective authority to use individual files, data items, programs and devices.
privacy	In connection of information systems privacy means that nobody can use the system or its parts without permission. On the other hand it means that the system or its owner cannot get data about the user without his permission.
HC interaction (HCI (human-computer interaction)	It is about the presentation of information on the screen and the means of controlling and operating the computer. Its about how the computer communicates with humans and vice versa.
support	Users need training and help when they use the system.
deployment	Deploying an information system means putting it in place so that it is ready to be used. The term refers mainly to delivering and installing hardware and software.
architecture and design of IS	Architecture is concerned with system's components or subsystems and relationships among these elements. It is a general level sketch of the system. It describes the coarse grain components of the system and covers both hardware and software. Detailed design, in turn, takes into consideration the platform on which the system is implemented. This means programming languages, operating systems, database management systems, etc.
maintenance	Maintenance means keeping information system running and in good condition. It includes regularly checking it and repairing when necessary.

(continues)

GLOSSARY OF DOMAINS USED IN WEBSITE CASE (continues)

DOMAIN	DEFINITION
change in IS	Every information system experiences unintended or intrinsic change (growth of database in size, deterioration, etc.). Sometimes it instigates maintenance operations. Secondly, change means intentional product evolution.
change in environment	The environment of an information system can be divided into at least three layers: user, user organization, society. All of these layers change constantly and cause pressure on developers to change the system itself.
change of environment	Sometimes an information system has to be moved to a different environment. In computer science this is called porting (adaptation of a piece of software so that it will function in a different computing environment to that for which it was originally written).
change of information system	Sooner or later all information systems must be replaced by new ones.
impact on environment	An information system may have impact on its user, user organization or the society as a whole. Influence can be positive or negative.
data, information, contents	This domain embraces simply all kinds of contents (database records, text, pictures, video, documents, etc.).
responsibility	Someone must be responsible of dealing with the information system, take decisions relating to it, take care of it, ...Resposibility concerns as well provider as user organization.
regulations, standards, conventions, laws	Some environmental factors affect the system even though the system does not rely on them directly. Intangible aspects of the context include explicit rules and requirements such as government regulations, industry standards, and organizational policies.
risk, harm to IS	Different things can cause physical harm to the information system. They include human actions as well as other factors like electricity surges. The harm may appear in the form of loss of data, lost network connection etc.
risk, harm to environment	This is specific and negative kind of impact on environment. The harm may meet users, user organizations, society, or the physical environment.
failure, fault, error in functionality	The information system does not succeed in doing or achieving something.
use of time and resources, performance	This domain could also be called efficiency, the ability to perform tasks without wasting time or other resources.

(continues)

GLOSSARY OF DOMAINS USED IN WEBSITE CASE (continues)

DOMAIN	DEFINITION
interaction with and integration to other ISs	Modern information systems usually act together with other information systems. This poses certain requirements to the parts of interaction.
sustainablity, suitability	These are composite attributes = they presuppose the existence of a number of other attributes. E.g. to be sustainable, a system must also be affordable, installable, supported, etc.

APPENDIX 7: GLOSSARY OF DOMAINS USED IN ZANZIBAR LAND REGISTRATION CASE

ACCESSIBILITY

"It is easy to obtain LRS, get into it and use it."

CAPACITY, SCALABILITY

Capacity and scalability refer to the system's capability to handle different amounts of data, services, service requests, and concurrent users, as such or, through configuration or by adding new constituents.

CHANGE OF PLATFORM

Sometimes an information system has to be moved to a different environment. In computer science this is called porting (adaptation of a piece of software so that it will function in a different computing environment to that for which it was originally written). ISO 9126 definition of portability: "capability of the software to be transferred from one environment to another".

CHANGE OF INFORMATION SYSTEM

Sooner or later all information systems must be replaced by new ones. ISO 9126 uses the term "replaceability": "capability of the software to be used in place of another specified software product for the same purpose in the same environment".

CONSUMPTION OF TIME AND RESOURCES, PERFORMANCE

Obtaining or building LRS, its maintenance and operation consume time and other resources (money, human resources, information, technology, other material resources). Performance, particularly, means the capability of system to provide appropriate response and processing times and throughput rates when performing its function (time behaviour in ISO 9126).

DATA, INFORMATION, CONTENTS

These terms refer to the information stored (permanently or temporally) or processed by the system.

DEVELOPMENT

The environment of an information system can be divided into four areas: infrastructure, user, user organization, society. All of these layers change constantly and cause pressure on developers to change or develop the system itself by adding new constituents and structures or altering existing ones.

DOCUMENTATION

Documentation refers to all kinds of source code and system descriptions, user and administration guides.

ERROR, FAILURE, RELIABILITY

Error means incorrect functioning of the system. It is differentiated from the incorrecness of contents. Failure means that the functioning of the system or some of its processes stops unexpectedly. Reliability means that LRS works without errors and failures, is available, accessible and integer for all time.

FEASIBILITY

Taking known requirements and the context into account it seems that the information system can be built.

HUMAN-COMPUTER INTERACTION

A wide area of concern. It is about the presentation of information on the screen and the means of controlling and operating the computer. Its about how the computer communicates with humans and vice versa. ISO 9126 uses the concept of usability: ”the capability of software product to be understood, learned, used and to be attractive to the user”. Nielsen (1993) defines usability as the question how well users can use the functionality offered by the system.

IMPACT ON ENVIRONMENT

An information system may have impact on its user, user organization or the society as a whole. Influence can be positive or negative.

INSTALLATION (DEPLOYMENT), OPERATION, MAINTENANCE

Operation means keeping LRS running. Maintenance means keeping LRS in good condition. It includes regularly checking it and repairing when necessary.

INTEGRATION TO OTHER SYSTEMS

Modern information systems usually need to act together with other information systems. This means, for example, exchange of data, messaging or sharing resources.

LAWS, REGULATIONS

”Capability of the software to adhere to laws, regulations.” (ISO 9126)

OWNERSHIP, RESPONSIBILITY

Ownership and responsibility refers to the willingness and motivation to own the information system and take care of it. Optimally the responsibility must be taken by both the system developer and user organizations.

SAFETY, INTEGRITY

”The capability of the software to achieve acceptable level of risk of harm to people, business, software, property or the environment in specified context of use.” (ISO 9126) “LRS is safe from physical failure, damage, loss, error, accidents etc. and consequences thereof.” In this context integrity means simply being whole and not having any missing or incoherent parts.

SECURITY

”Capability of the software to protect information and data so that unauthorized persons or systems cannot read or modify them and authorized persons or systems are not denied access to them.” (ISO 9126) Capability to protect hardware, software and data ....

SERVICES

Services are all the things the system can do with information for the users. Functioning of the software and hardware is used to produce these services. ISO 9126 uses the term functionality: ”the capacity of the software to provide functionswhich meet stated and implied needs”. Services can be work or entertainment (e.g. games, music) oriented.

SOFTWARE AND HARDWARE PLATFORM

Platform includes hardware, operating systems, programming languages, application frameworks, runtime libraries, application servers, DBMSs, networking components, etc. that are used as tools or elements in the system development. They are, however, not themselves designed and build in the system development process.

SOURCE CODE

Source code is the collection of files written by the programmers in some human readable programming language. It is then converted into computer-executable form.

STANDARDS, CONVENTIONS

"Capability of the software to adhere to standards and conventions." (ISO 9126)

SUPPORT, TRAINING

Support means the help users, operators, administrators etc. get from inside or outside of the organization using the information system (LRS).

SUPPORT TO BUSINESS PROCESSES

Business activities and processes produce something of value to an organization and its customers. An information system, like LRS, must assist in conducting these activities and business processes.

SUSTAINABILITY

"LRS is wanted and can be maintained by the user organizations and kept functioning at the same rate or level without any problems over a desired period of time."

USEFULNESS

"... wether the system can be used to achieve some desired goal." (Nielsen 1993) "LRS can help you to achieve something in a particular situation."

APPENDIX 8: PRIORITIES GIVEN TO DOMAINS IN EMIS CASE

DOMAIN	Inverse Average	Average	Actor 1	2	3	4	5	6	7	8	9	10	11
support to business processes of user organization	13.09	1.91	1	1	1	1	3	2	1	2	3	5	1
authorization	8.18	6.82	3	3	9	5	4	8	7	9	12	11	4
HC interaction	10.64	4.36	4	3	6	4	1	3	6	8	4	7	2
deployment	9.09	5.91	5	4	2	2	10	15	2	6	6	4	9
architecture and design of IS	5.36	9.64	5	15	15	3	14	15	15	14	2	1	7
maintenance	6.09	8.91	5	15	7	6	11	10	15	13	7	3	6
change in IS	3.27	11.73	5	15	15	8	15	15	15	12	9	15	5
change in environment	5.00	10.00	5	15	15	7	12	6	4	10	15	13	8
change of environment	3.00	12.00	5	15	15	9	13	15	3	11	16	15	15
change of IS	15.00	0.00											
impact on environment, influence	4.55	10.45	9	8	8	10	15	9	10	5	17	9	15
data, information, contents	9.82	5.18	7	2	3	11	2	1	15	1	1	2	12
regulation, standard, convention, law	4.45	10.55	6	10	10	12	15	7	15	3	13	10	15
risk, harm to the IS itself	3.27	11.73	9	9	15	14	7	15	15	16	10	16	3
risk, harm to environment	2.18	12.82	9	15	15	13	8	15	8	15	14	14	15
failure, fault, error in functionality	2.73	12.27	8	5	15	15	9	15	15	17	11	12	13
use or time and resources, performance	7.09	7.91	8	7	4	15	5	5	9	7	8	8	11
interaction with and integration to other ISs	8.82	6.18	2	6	5	15	6	4	5	4	5	6	10
composite attributes	0.00	15.00	15	15	15	15	15	15	15	15	15	15	15

APPENDIX 9: PRIORITIES GIVEN TO DOMAINS IN WEBSITE CASE

DOMAIN	Inverse Average	Average	Actor 1	2	3	4	5
usage	2.80	12.20	15	15	1	15	15
support to business processes of user organization	12.40	2.60	7	1	2	2	1
entertainment	4.00	11.00	6	4	15	15	15
authorization	4.80	10.20	15	8	10	15	3
privacy	1.20	13.80	15	15	9	15	15
HC interaction	4.80	10.20	15	3	15	3	15
support	6.40	8.60	9	15	6	7	6
deployment	4.00	11.00	15	11	7	15	7
architecture and design of IS	6.80	8.20	3	7	15	1	15
maintenance	8.40	6.60	8	9	8	4	4
change in IS	0.40	14.60	15	15	15	13	15
change in environment	3.60	11.40	4	14	15	16	8
change of environment	0.60	14.40	15	12	15	15	15
change of IS	0.60	14.40	15	13	15	14	15
impact on environment, influence	3.00	12.00	15	6	15	15	9
data, information, contents	9.20	5.80	1	5	3	5	15
responsibility	8.80	6.20	5	10	5	6	5
regulation, standard, convention, law	0.00	15.00	15	15	15	15	15
risk, harm to the IS itself	3.20	11.80	11	15	15	8	10
risk, harm to environment	1.20	13.80	15	15	15	9	15
failure, fault, error in functionality	1.00	14.00	15	15	15	10	15
use of time and resources, performance	8.60	6.40	2	2	15	11	2
interaction with and integration to other ISs	3.80	11.20	10	15	4	12	15
composite attributes	0.00	15.00	15	15	15	15	15

APPENDIX 10: PRIORITIES GIVEN TO ATTRIBUTES IN EMIS CASE

ATTRIBUTE	Inverse Average	Average	Actor 1	2	3	4	5	6	7	8	9	10	11
being needed	4.27	10.73	5	1	15	15	15	15	15	15	6	15	1
comprehensiveness	1.27	13.73	15	15	15	15	15	15	6	15	15	10	15
appropriateness	0.55	14.45	15	15	15	15	15	15	15	9	15	15	15
timeliness	1.09	13.91	15	15	6	15	15	15	15	15	15	12	15
coordination	1.45	13.55	15	15	15	15	10	15	15	15	8	11	15
type and degree of use	0.55	14.45	15	15	15	15	15	15	15	15	9	15	15
usefulness	2.73	12.27	15	15	15	15	2	7	15	8	15	13	15
regularity	0.45	14.55	15	15	15	15	15	10	15	15	15	15	15
security	0.82	14.18	15	6	15	15	15	15	15	15	15	15	15
clearness	1.09	13.91	15	15	15	15	3	15	15	15	15	15	15
ease of use	2.55	12.45	15	15	5	15	15	15	5	15	7	15	15
user orientation	2.27	12.73	4	15	15	15	1	15	15	15	15	15	15
user friendliness	6.36	8.64	15	3	2	15	15	6	15	1	15	6	2
accessibility	2.27	12.73	15	10	15	15	15	15	15	15	15	7	3
simplicity	2.18	12.82	15	15	4	15	15	15	15	15	8	9	15
availability	3.09	11.91	15	15	15	5	6	15	7	15	15	8	15
installability	5.82	9.18	6	4	15	3	7	15	15	15	15	1	5
type and degree of impact on environment	3.45	11.55	7	15	7	9	15	11	15	3	15	15	15

(continues)

PRIORITIES GIVEN TO ATTRIBUTES IN EMIS CASE (continues)

ATTRIBUTE	Inverse Average	Average	Actor 1	2	3	4	5	6	7	8	9	10	11
coverage	4.73	10.27	15	15	15	15	15	2	8	15	4	2	7
reliability	2.91	12.09	15	15	2	15	8	4	15	15	15	14	15
validity	3.36	11.64	15	8	15	6	15	15	15	7	2	15	15
accuracy	4.64	10.36	15	9	15	15	4	3	2	15	15	15	6
adequacy	0.55	14.45	15	15	15	15	15	9	15	15	15	15	15
correctness	1.82	13.18	15	15	15	15	15	15	9	15	1	15	15
relevancy	2.55	12.45	15	11	9	8	15	15	15	4	15	15	15
non-overlap	1.09	13.91	15	15	15	15	15	15	15	15	3	15	15
being up-to-date	4.00	11.00	15	15	15	15	15	5	10	2	5	15	9
being needed	15.00	0.00											
compliance	1.00	14.00	15	15	15	4	15	15	15	15	15	15	15
security	5.09	9.91	15	5	8	7	5	15	15	5	15	15	4
environment friendliness	1.55	13.45	15	15	15	10	15	15	3	15	15	15	15
reliability	6.36	8.64	2	2	3	2	15	15	15	15	11	5	10
affordability	15.00	0.00											
compatibility	5.09	9.91	3	15	10	15	9	15	15	6	10	3	8
sharing	4.45	10.55	3	7	15	15	15	8	4	15	15	4	15
sustainability	3.64	11.36	8	15	15	1	15	1	15	10	15	15	15
suitability	3.82	11.18	1	1	15	15	15	15	1	15	15	15	15

APPENDIX 11: PRIORITIES GIVEN TO ATTRIBUTES IN WEBSITE CASE

ATTRIBUTE	Inverse Average	Average	Actor 1	2	3	4	5
type and degree of use	1.60	13.40	15	15	7	15	15
being needed	0.00	15.00	15	15	15	15	15
comprehensiveness	0.00	15.00	15	15	15	15	15
appropriateness	0.00	15.00	15	15	15	15	15
timeliness	0.00	15.00	15	15	15	15	15
coordination	0.00	15.00	15	15	15	15	15
type and degree of use	0.00	15.00	15	15	15	15	15
usefulness	4.80	10.20	15	7	7	7	15
regularity	0.00	15.00	15	15	15	15	15
security	3.20	11.80	15	7	15	15	7
privacy	1.60	13.40	15	15	7	15	15
clearness	0.00	15.00	15	15	15	15	15
ease of learning	1.60	13.40	15	7	15	15	15
ease of use	3.20	11.80	15	7	7	15	15
user orientation	0.00	15.00	15	15	15	15	15
user friendliness	1.60	13.40	15	15	7	15	15
accessibility	1.60	13.40	15	15	7	15	15
accessibility 2	1.60	13.40	15	15	7	15	15
simplicity	1.60	13.40	15	15	7	15	15
availability	1.60	13.40	15	15	7	15	15
attractiveness	3.20	11.80	15	7	15	7	15
installability	0.00	15.00	15	15	15	15	15

(continues)

PRIORITIES GIVEN TO ATTRIBUTES IN WEBSITE CASE (continues)

ATTRIBUTE	Inverse Average	Average	Actor 1	2	3	4	5
ease of deployment	1.60	13.40	15	15	7	15	15
commonness	1.60	13.40	15	7	15	15	15
availability	1.60	13.40	15	7	15	15	15
(in)expensiveness	1.60	13.40	15	7	15	15	15
ease of maintenance	1.60	13.40	15	15	7	15	15
amount of maintenance work	1.60	13.40	15	7	15	15	15
regularity	1.60	13.40	15	15	15	15	7
ease of repairing	1.60	13.40	15	7	15	15	15
ease of updating	3.20	11.80	15	7	15	7	15
(in)dependence (from) on environment	1.60	13.40	15	7	15	15	15
adaptability	1.60	13.40	15	7	15	15	15
ease of migration	1.60	13.40	15	7	15	15	15
type and degree of impact on environment	1.60	13.40	15	15	15	7	15
financial impact on host organization	1.60	13.40	15	7	15	15	15
impact on user	1.60	13.40	15	15	15	15	7
coverage	0.00	15.00	15	15	15	15	15
reliability	1.60	13.40	15	15	7	15	15
validity	0.00	15.00	15	15	15	15	15
accuracy	0.00	15.00	15	15	15	15	15
adequacy	0.00	15.00	15	15	15	15	15
correctness	0.00	15.00	15	15	15	15	15
relevancy	0.00	15.00	15	15	15	15	15

(continues)

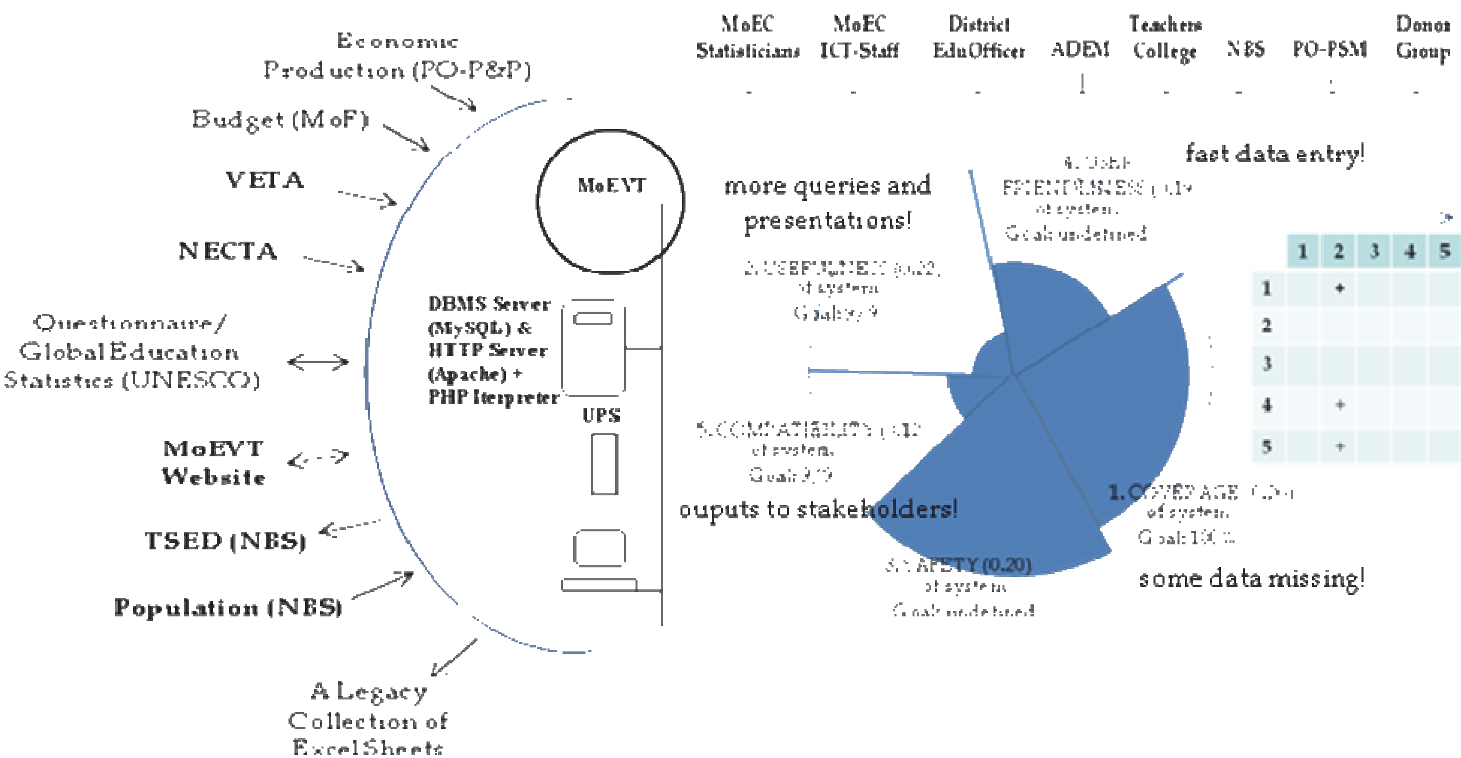
PRIORITIES GIVEN TO ATTRIBUTES IN WEBSITE CASE (continues)

ATTRIBUTE	Inverse Average	Average	Actor 1	2	3	4	5
non-overlap	0.00	15.00	15	15	15	15	15
being up-to-date	4.80	10.20	7	15	7	7	15
ease of contacting those who are responsible	1.60	13.40	15	15	15	7	15
degree of definition	3.20	11.80	15	7	15	15	7
way of handling	1.60	13.40	15	15	7	15	15
flexibility	1.60	13.40	15	15	7	15	15
being supported	3.20	11.80	15	15	7	15	7
compliance	0.00	15.00	15	15	15	15	15
being certified	1.60	13.40	15	15	15	7	15
security	1.60	13.40	15	15	15	15	7
environment friendliness	0.00	15.00	15	15	15	15	15
reliability	0.00	15.00	15	15	15	15	15
execution speed	1.60	13.40	15	15	15	7	15
cost-effectiveness	1.60	13.40	15	15	7	15	15
efficiency	1.60	13.40	15	15	15	15	7
compatibility	0.00	15.00	15	15	15	15	15
browser compatibility	3.20	11.80	15	15	7	7	15
sharing	0.00	15.00	15	15	15	15	15
sustainability	0.00	15.00	15	15	15	15	15
suitability	0.00	15.00	15	15	15	15	15

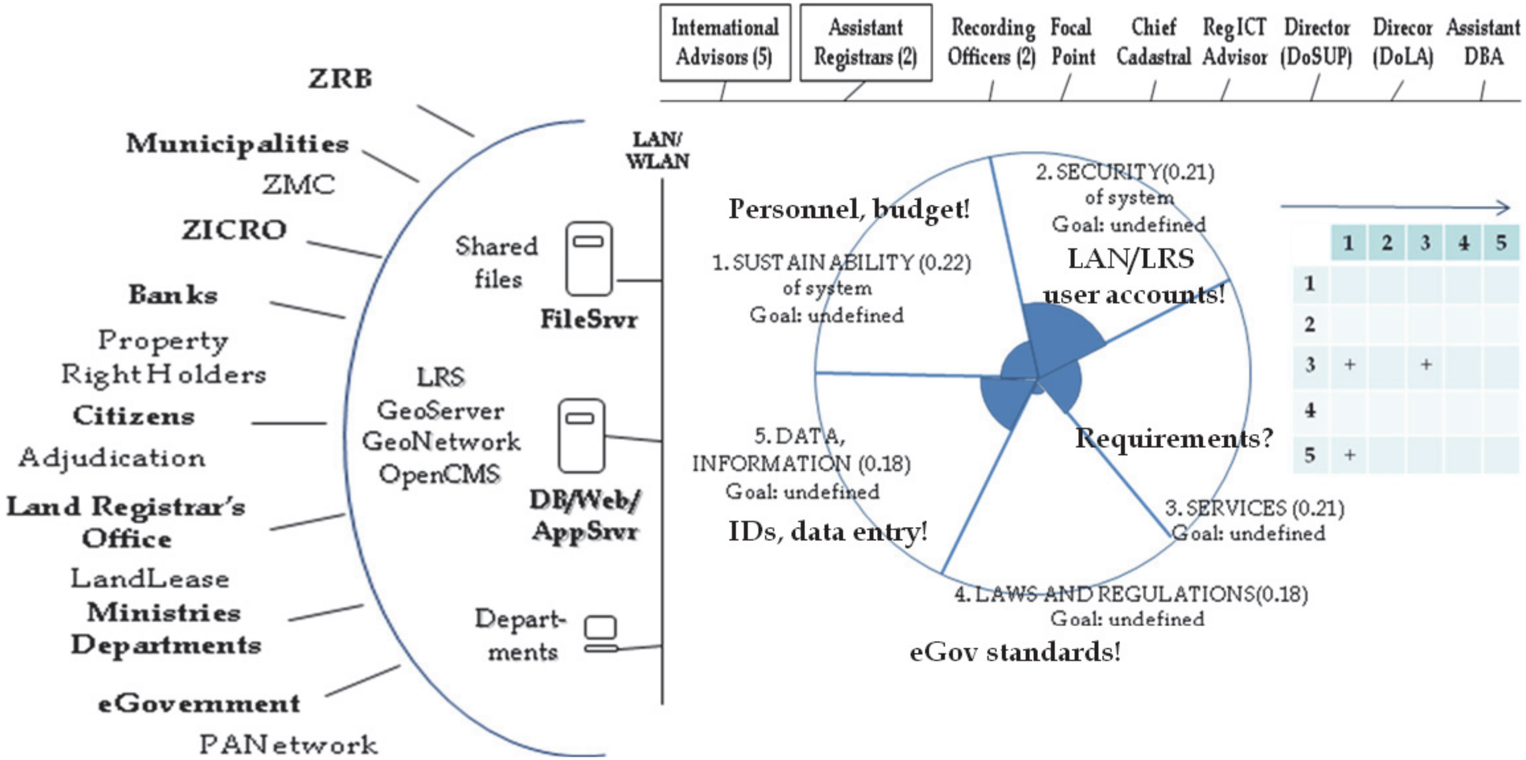
APPENDIX 12: PRIORITIES GIVEN TO DOMAINS IN LRS CASE

DOMAIN	Inverse Average	Average	Actor1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
sustainability	8.27	6.73	1	2	15	15	1	13	3	1	7	8	3	12	2	9	9
availability, accessibility	5.53	9.47	15	15	4	15	15	7	5	6	2	15	2	10	15	10	6
safety, integrity	5.20	9.80	15	9	15	10	3	5	1	15	11	3	15	7	15	8	15
security	8.07	6.93	6	15	15	1	4	1	2	7	8	11	4	6	15	4	5
errors, failures, reliability	1.67	13.33	15	15	15	15	15	9	7	10	15	13	15	15	15	11	15
services	7.93	7.07	3	7	11	6	15	3	6	15	15	2	10	8	3	1	1
support to business processes	5.27	9.73	2	8	15	15	15	11	15	15	12	1	1	15	1	5	15
human-computer interaction	1.67	13.33	13	15	15	15	15	15	15	15	15	14	15	3	5	15	15
impact on environment	1.13	13.87	15	13	15	15	8	15	15	15	15	7	15	15	15	15	15
integration to other systems	2.60	12.40	4	15	3	15	5	15	15	15	9	15	15	15	15	15	15
laws, regulations	7.00	8.00	7	1	7	9	10	2	15	3	15	6	9	4	15	15	2
standards, conventions	1.73	13.27	15	10	12	15	15	8	10	9	15	15	15	15	15	15	15
data, information, contents	6.80	8.20	12	15	13	7	15	6	9	15	1	5	11	1	4	2	7
source code	1.33	13.67	15	15	15	4	15	15	11	15	15	10	15	15	15	15	15
documentation	5.53	9.47	9	5	9	2	15	10	8	15	6	15	5	15	9	15	4
support, training	5.00	10.00	15	4	6	5	2	14	4	8	10	15	7	15	15	15	15
ownership, responsibility	5.93	9.07	5	6	15	3	6	12	13	15	4	9	15	5	15	3	10
administration	6.67	8.33	8	3	1	15	7	15	14	2	3	4	8	15	7	15	8
change of platform	0.47	14.53	15	15	8	15	15	15	15	15	15	15	15	15	15	15	15
change of information system	0.80	14.20	15	15	10	8	15	15	15	15	15	15	15	15	15	15	15
installation, operation, maintenance	4.33	10.67	10	14	2	15	15	15	12	5	5	15	15	9	6	7	15
further development	1.80	13.20	15	12	15	15	15	15	15	11	15	15	15	2	8	15	15
consumption of time and resources, performance	2.40	12.60	15	11	15	15	15	4	15	15	15	12	6	15	15	6	15
capacity, scalability	3.13	11.87	11	15	5	15	9	15	15	4	15	15	15	11	15	15	3

APPENDIX 13: POSTER OF EMIS QUALITY MODEL

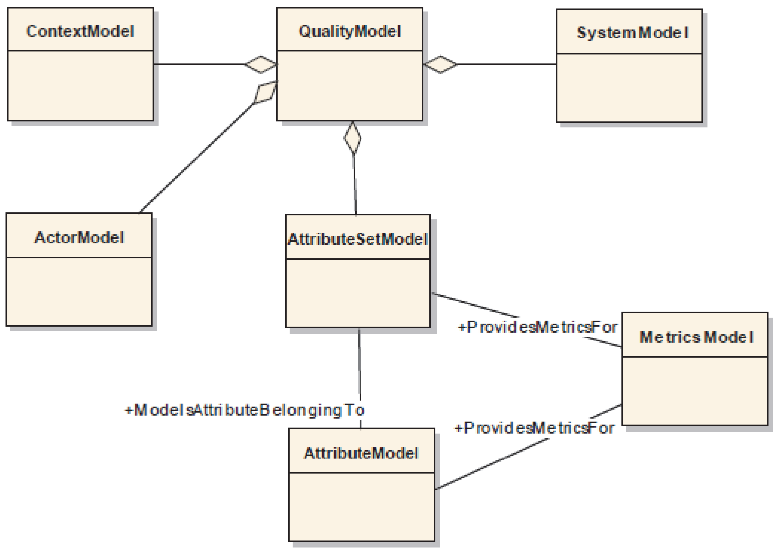


APPENDIX 14: POSTER OF LRS QUALITY MODEL

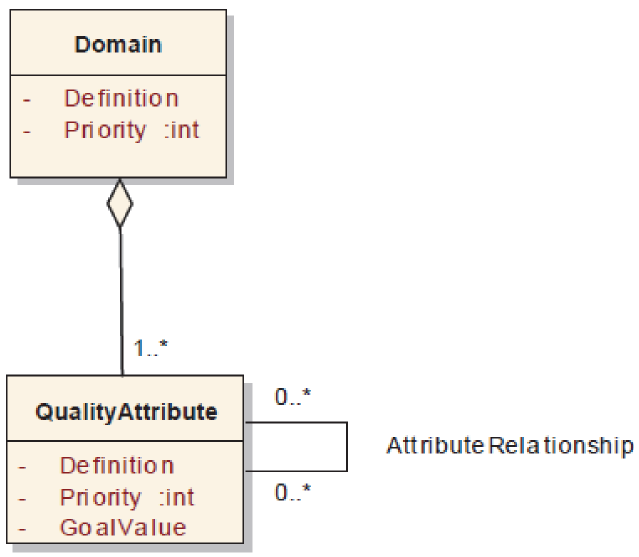


	1	2	3	4	5
1					
2					
3	+		+		
4					
5	+				

APPENDIX 15: OVERALL VIEW OF META-MODEL AND ATTRIBUTE SET MODEL AS UML-DIAGRAMS

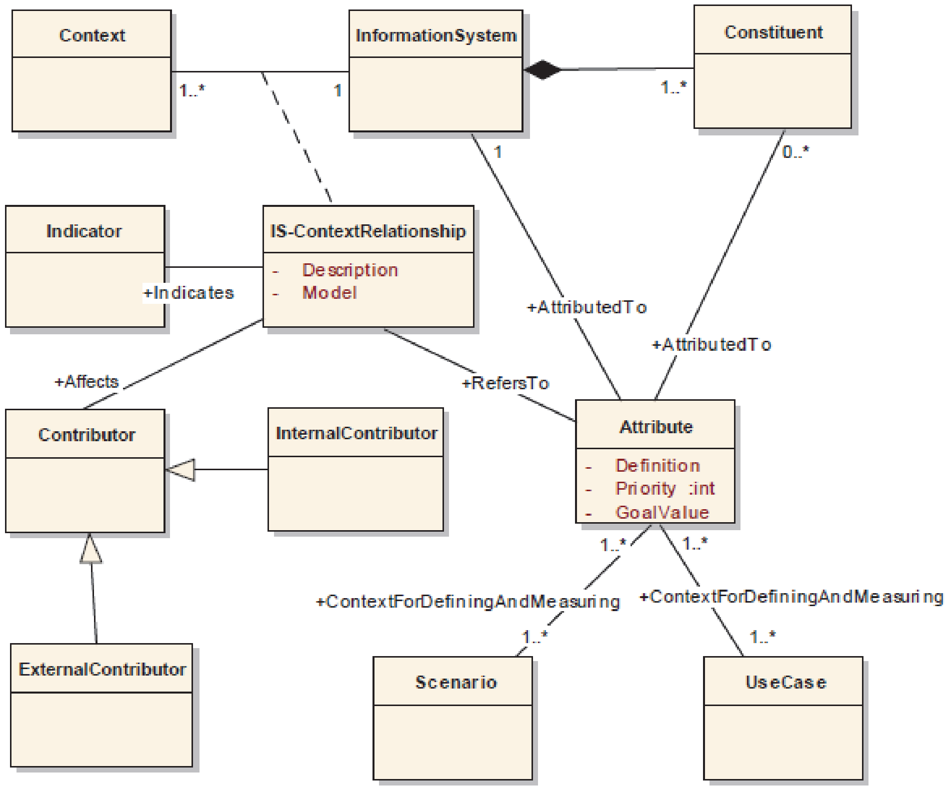


Overall view of information system quality meta-model



Attribute set model

APPENDIX 16: ATTRIBUTE MODEL AS UML-DIAGRAM



Attribute model