Extending the Concept of Quality in Systems Development — Integrating Software and Information Quality

Anna Wingkvist[1], Morgan Ericsson[2], Welf Löwe[1], Rüdiger Lincke[1]

[1]Linnaeus University, [2]Uppsala University

anna.wingkvist@lnu.se, morgan.ericsson@it.uu.se, welf.lowe@lnu.se, rudiger.lincke@lnu.se

Abstract. The usefulness and value of an information system is directly related to its perceived quality. Quality is multidimensional concept, and includes an object of interest, the viewpoint on that object and the qualities attributed to the object. This suggests that there is no universal standard in systems development; quality is rather defined how well the information system meets the purpose and the goals of the organization it is used within. It is important that people involved in a particular systems development project have an agreed understanding of what the strive for quality means. This agreed understanding should include how to assign appropriate quality characteristics to both the technical and social aspects of a system as well as how to assess and interpret them. The purpose of this paper is twofold; first, we emphasize that any definition of quality should be specific to a system, and include both the social and technical aspects of a system. Second, we extend methods used to define and assess quality to include social and technical aspects that extends beyond software. Our work is particularly focused on information quality.

Key words: Systems development, Socio-technical systems, Information quality, Quality models.

1 Introduction

The usefulness and value of an information system depends on a number of factors, such as reliability, efficiency, flexibility, and so on. Factors such as these are often regarded characteristics related to the *quality* of an information system.

Quality can be considered a multidimensional concept. There is an object of interest, in this case the information system. The object is considered from a point of view, and evaluated according to specific characteristics, such as efficiency. The value of such characteristics depends on the point of view.

We can establish that quality is measured according to certain characteristics. These characteristics, how they relate to each other, and how they are measured are formalized as *quality models*. These quality models related to the entire information system, and can be considered to cover both social and technical aspects.

The quality of information in an information system is an important and social aspect. Information Quality (IQ) is defined as *"fitness for use"* of the information provided and is basically a measure of the use it provides to the users of that information. The measure of IQ is to a large extent subjective. There exist several models and frameworks to help define and assess information quality (cf. Section 2).

However, when measuring the quality of an information system, there is a focus on the software and software development process. Quality is assessed for the produced information system, and the documents produced as part of the development process. While socio-technical aspects, such as usability and security, are considered, the social aspects, such as information, are not.

In this paper we investigate how a quality model that considers social aspects, information in particular, can be established. We begin by discussing how quality is defined, with a particular focus on software and information in Section 2. In Section 3 we analyze these definitions and establish a method to define quality models that capture both software and information characteristics. Section 4 concludes the paper and presents future work.

2 Quality

In this section we investigate the definitions of quality and ways to assess it. We begin by discussing quality in general and how it can be defined. We then discuss various definitions of quality with respect to software and information.

The notion of quality is used in daily language as an intangible trait, something that can be subjectively felt or judged, but often not exactly measured or weighted. Terms like good or bad quality are intently vague and used with no intention of ever being an exact science. Another issue is that quality is multidimensional, and includes an object of interest, the viewpoint on that object and the quality attributes attributed to the object. This can make *"quality"* a confusing concept.

In order to discuss quality, and to be able to assess and improve it, it has to be defined. Crosby (1979) defines quality as *"conformance to requirements"*. This definition suggests that there must exist a set of requirements that are defined in such a way that they cannot be misunderstood. Something that is of quality conforms to these requirements and anything that does not conform is considered a defect. The requirements are not to be considered universal, but set by an entity or for a single product.

Another notion of quality is given by Juran (1998) who defines quality as *"fitness for use"*. This definition considers the users, and the requirements and expectations that they have on the product and their use of it. Juran further states that since different users may use the product in different ways, the product must possess multiple elements of fitness of use. These elements are quality characteristics that can be divided into parameters.

The two definitions of quality can seem unrelated, but complement each other in reality. The demands and expectations of the users guide the requirements set by the producers. So, a conformance to the well motivated requirements generally means that the product will be fit for use as well.

2.1 Software Quality

McCall et al. (1977) present a quality model defining 11 quality factors that relate to the three stages of a simplified software life cycle: revision, operation, and transition. McCall et al. also define about 22 metrics that are used to measure the quality of the factors. Several metrics are weighted and used to determine the quality of each factor. Many of the metrics are based on

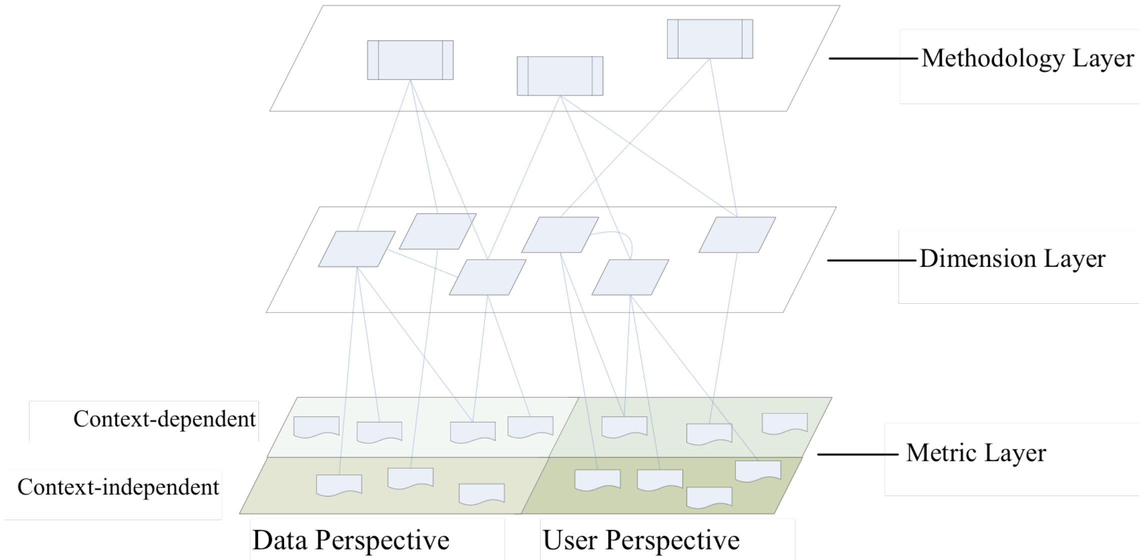

Figure 1: An information quality framework

checklists and a 0 to 10 scale, which means that they are subjectively measured. This quality model is standardized in ISO/IEC 9126-1:2001 (ISO, 2001). Quality models also differentiate perspectives, stages or phases. There are several aspects of quality and not all of these may be appropriate for all perspectives and phases. McConnell (2004) differentiate between internal and external quality, i.e., quality that affects the product while produced versus quality when the product is in use. These aspects are standardized in ISO/IEC 9126, as well.

The model by McCall et al. introduces several important ideas. First, there is not one product quality, but several factors that affect the product quality. Second, these factors matter during different periods of the life cycle. Third, the quality factors should be measurable and metrics should be defined. Several modern quality models and metrics suites exist, such as those by Li and Henry (1993), Chidamber and Kemerer (1994), Abreu (1995), and Henderson-Sellers (1996). There are several studies that validate the claim that metrics can be used as an indicator of the quality of software, for example Basili et al. (1996) and Harrison et al. (1998).

2.2 Information Quality

In regard to information as the object of investigation, Wang and Strong (1996) (building on Juran (1998)) define Information Quality *"as fitness for use by information users"*. Although, Klein et al. (1997) showed that information users had trouble pointing out and detecting errors in information and altering the way they use it. Hence, placing all the responsibility on a user is not an appropriate approach. In response to this, Kahn et al. (2002) suggested a view where quality also depends on conforming to specifications or requirements (adopted from Crosby (1979)).

As Information Quality as a research topic is gaining attention, the need to reach common ground on assessment is evident. There have been a number of suggestions on the approach and classification of IQ assessment, e.g., Bovee et al. (2003), Helfert (2001), Kahn et al. (2002), Naumann and Rolker (2000), Wang and Strong (1996), Wand and Wang (1996). The basis of all of these researchers is that IQ assessment can be defined as the process of assigning numerical or categorical values to information in a given setting. Ge and Helfert (2007) reviewed the literature related to IQ assessment and were able to organize IQ assessment into three layers; methodology, dimension, and metric. The methodology layer includes IQ assessment models, frameworks and methodologies, i.e., a dimension specification from both a data and user perspective. The dimension layer in turn determines the characteristics of the information i.e., presenting the categories to measure IQ from intuitive, theoretical, and empirical stance. The metric layer is represented by a two by two conceptual model which range from data perspective to user perspective, context-dependent to context-independent, i.e., a classification to control IQ by automated and use case processes respectively.

3 Integrating Information Quality and Software Quality

Quality is defined by and assessed according to established quality models. In order to include social aspects, such as information, these need to be included in the quality model used during the development process. In this section we establish a quality model that includes both software and information.

In Section 2 we investigate how quality models are established for both software and information. While there are several similarities between the models and methods used, there are also differences. Consider quality characteristics such as reliability. On a high and intuitive level, the meaning is the same, but on a practical level the definitions and means to assess differ greatly.

A new quality model should be established based on the previous, but not by combining models for the two fields. In this section we first analyze the models from the two fields and establish a set of requirements on a *"common quality model"*. Based on these requirements, we propose an approach to create quality models.

3.1 Analysis of Software and Information Quality

Both the software and information quality communities agree that quality can be measured. In this sense, both are close to the Crosby's definition of quality. Quality is measured using metrics. A metric is a measurement of a property, for example the number of (text) clones in the text or source code. The use and definitions of metrics are similar in software and information quality; we have for example studied how metrics from software quality assessment can be adapted for information quality (Wingkvist, Löwe, Ericsson and Lincke, 2010).

The metrics are abstracted and grouped into factors, dimensions, phases and so on. For example, McCall et al. (1977) group metrics into factors, and the factors are in turned group by which phase of the systems life cycle the factor applies to. How the metrics are grouped, and what metrics are related to which factors differ greatly between software and information quality. As discussed in Section 2 it is common to different between context-dependent and context-independent as well as user and data metrics in information quality, for example.

The quality models are generally defined using a bottom-up perspective, focusing on what can be measured. Different factors that approximate what can be considered quality are defined, and then grouped into hierarchies. The bottom-up perspective can be problematic for several reasons. One problem is that there is a risk of a metrics explosion, since many things are possible to measure, but not everything is relevant. A related but opposite problem is that sometimes what is quick and easy to measure is not the only thing that should be considered; there is a risk socio-technical and social concerns are disregarded since these can be difficult to measure.

Another issue that originates from the bottom-up perspective is that quality is defined in terms of the characteristics. For example, McCall et al. defines, among others, usability, efficiency, and flexibility, as quality characteristics, and these will be synonymous with quality when using a bottom-up perspective. Given that there exist positive and negative correlations (relationships) (Ge and Helfert, 2007) between such quality characteristics the notion of quality gets more complicated.

For example, Efficiency and Flexibility are generally not separate entities; increasing one may decrease the other. It is not possible to have high quality by getting good measurements for all characteristics; quality must rather be defined as a profile.

3.2 Requirements on a Common Model

Measurements and metrics are can be considered the foundation of quality, both for information and software, and any common quality model should support these. The way the metrics are abstracted and grouped into factors and such is problematic for several reasons. The definitions and relationships between the factors is already complicated, and when combining existing models, there is a risk the number of characteristics will increase significantly. While a quality profile using a small number of characteristics is feasible, it will become infeasible when the number of characteristics grow. We instead propose that each organization, information system, and context define quality goals in a top-down fashion, and use the appropriate characteristics to support these quality goals. This will increase the need for metrics that measure socio-technical or social aspects.

In order for the model to be used, it should support any systems development methodology and life cycle model. It would be beneficial if the model was based on existing approaches and methods already known and accepted by the software engineering and systems development communities.

We formulate the following five requirements on a common quality model:

1. Support measurement and quality metrics,
2. Use a top-down perspective,
3. Allow for socio-technical metrics,
4. Support quality goals set by the organization for the information system and context,
5. Be neutral to systems development methodologies and life cycle models.

3.3 An Approach to Creating Quality Models

The requirements that the common quality model should use a top-down perspective and allow the organizations to set their own quality goals, it is not feasible to find a single model. Goal-Question-Metric (GQM) (Basili et al., 1994) is a goal-driven approach to (software) measurement. The goals are specified by an organization or a project, and are central in order to create a purposeful way to measure. The goals are traced to data that is intended to define the goals, and the data is interpreted with respect to the framework. GQM is not a measurement or quality model, but an approach to produce such models. When we apply GQM we get a measurement model that consists of a measurement system specific to our issues and rules that help interpret the measured data. The produced measurement model consists of three levels: conceptual, operational, and quantitive.

The conceptual level contains the goals. A goal is defined for an object, such as a product, a process, or a resource, and is an overarching property that the object should posses. The goals can be defined from various points of views, relative to particular environments, for specific reasons, etc. A goal is often specified in terms of purpose, issue, object and viewpoint, for example *"Make information easily accessible from an user perspective"*.

The operational level is concerned with questions. Questions are used to characterize how specific goals can be achieved or assessed. The questions breaks down a goal into its major components, in order to try to characterize the object of study with respect to specific quality issues. For example, a question supporting the previous goal could be *"Is it easy to find the information* *you are looking for?"*. The questions are broken down into metrics on the quantitive level. The metrics query data in order to answer the questions. The data queried can be either objective or subjective. The question specified can use subjective data from a questionnaire where employees were asked to rate how easy it is to find information on a Likert scale, for example.

3.4 Evaluation of the Suggested Approach

Goal-Question-Metric is a top-down approach to create specific measurement models. In this sense, it conforms to the first two requirements set. The specific measurement model depends on goal defined for an organization or a project that can reflect particular perspectives and situations. So, GQM conforms to the third requirement as well. The metrics and data that the goals are supported by can be subjective or objective. The subjective data can capture user satisfaction, readability or other concerns where users interact with the system. Subjective and objective data covers the socio-technical aspect. According to Basili et al. (1994), GQM was designed to be applicable to all life cycle products, processes, and resources.

4 Conclusion

The quality of the software is not the only aspect of quality that matters for an information system. In this paper, we study how to extend the quality assessment and assurance processes during software development and maintenance to also include what we refer to as social aspects, such as information. Our aim is to provide a more complete notion of quality.

In order to establish a more complete notion of quality, we investigated how quality is defined and assessed for both software and information perspectives. We find that the existing models often have a bottom-up perspective, that becomes problematic. We instead suggest a top-down approached based on the Goal-Question-Metric that allows organizations to set and assess their own quality goals depending on, for example, the information system and its use.

In this paper the main focus is on motivating and establishing our approach. We have also conducted limited experiments, were we assessed the quality of technical documentation (Wingkvist, Ericsson, Löwe and Lincke, 2010) with promising results. We intend to continue to evaluate and improve the approach to quality suggest in this paper, both in theory and practice.

We have currently focused on quality characteristics that deal either with information or software. It will be very interesting to see how characteristics that focus on both can be formulated and assessed. For example, the reliability of an information system depends both on the reliability of the software and information. We are also interested in finding good ways to communicate the quality, for example using different visualizations, suitable for different stakeholders.

Acknowledgment

Anna Wingkvist is supported by the Swedish Research School of Management and IT (MIT), Uppsala, while Morgan Ericsson is supported by the Uppsala Programming for Multicore Architectures Research Center (UPMARC). Also, we would like to thank the Knowledge Foundation for financing part of the research with the project, "Validation of metric-based quality control", 2005/0218.

References

Abreu, F. B. (1995), The MOOD metrics set, in 'Proceedings ECOOP Workshop on Metrics'.

Basili, V. R., Briand, L. C. and Melo, W. L. (1996), 'A Validation of Object-Oriented Design Metrics as Quality Indicators', IEEE Transactions on Software Engineering 22(10), 751–761.

Basili, V. R., Caldiera, G. and Rombach, H. D. (1994), The goal question metric approach, in 'Encyclopedia of Software Engineering', Wiley.

Bovee, M., Srivastava, R. and Mak, B. (2003), 'A conceptual framework and belief-function approach to assessing overall information quality', International journal of intelligent systems 18(1), 51–74.

Chidamber, S. R. and Kemerer, C. F. (1994), 'A Metrics Suite for Object-Oriented Design', IEEE Transactions on Software Engineering 20(6), 476– 493.

Crosby, P. B. (1979), Quality is free: the art of making quality certain, McGraw-Hill, New York.

Ge, M. and Helfert, M. (2007), A review of information quality research — develop a research agenda, in 'Proceedings of the 12th International Conference on Information Quality'.

Harrison, R., Counsell, S. J. and Nithi, R. V. (1998), 'An Investigation into the Applicability and Validity of Object-Oriented Design Metrics', Empirical Software Engineering 3(3), 255–273.

Helfert, M. (2001), Managing and measuring data quality in data warehousing, in 'Proceedings of the World Multiconference on Systemics, Cybernetics and Informatics', pp. 55–65.

Henderson-Sellers, B. (1996), Object-oriented metrics: measures of complexity, Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

ISO (2001), 'ISO/IEC 9126-1 "Software engineering - Product Quality - Part 1: Quality model".

Juran, J. (1998), Juran's Quality Control Handbook, 5th edn, McGraw-Hill.

Kahn, B. K., Strong, D. M. and Wang, R. Y. (2002), 'Information quality benchmarks: product and service performance', Commun. ACM 45(4), 184– 192.

Klein, B. D., Goodhue, D. L. and Davis, G. B. (1997), 'Can humans detect errors in data? Impact of base rates, incentives, and goals', MIS Q. 21(2), 169–194.

Li, W. and Henry, S. (1993), Maintenance metrics for the object oriented paradigm, in 'IEEE Proceedings of the First International Software Metrics Symposium', pp. 52–60.

McCall, J. A., Richards, P. G. and Walters, G. F. (1977), Factors in Software Quality, Technical Report Volume I. NTIS AD/A-049 014.

McConnell, S. (2004), Code Complete, second edn, Microsoft Press, Redmond, WA, USA.

Naumann, F. and Rolker, C. (2000), Assessment methods for information quality criteria, in 'Proceedings of the fifth International Conference on Information Quality', pp. 148–162.

Wand, Y. and Wang, R. Y. (1996), 'Anchoring data quality dimensions in ontological foundations', Communications of the ACM 39(11), 86–95.

Wang, R. Y. and Strong, D. M. (1996), 'Beyond accuracy: what data quality means to data consumers', Journal of Management Information Systems 12(4), 5–33.

Wingkvist, A., Ericsson, M., Löwe, W. and Lincke, R. (2010), A metrics-based approach to technical documentation quality, in 'Proceedings of the 7th International Conference on the Quality of Information and Communications Technology'. Forthcoming.

Wingkvist, A., Löwe, W., Ericsson, M. and Lincke, R. (2010), Analysis and visualization of information quality of technical documentation, in '4th European Conference on Information Management and Evaluation'. Forthcoming.