Influencing Qualities of Information Systems – Future Challenges for Participatory Design

Kristin Braa
Department of Informatics
University of Oslo, POBox 1080 Blindern,
N-0316 Oslo, Norway
email: kbraa@ifi.uio.no

ABSTRACT
This paper aims at expanding the approach to IS quality – beyond the formal technical perspective, to include use and organizational perspectives, as well as calling for recognition from the Participatory Design community to intervene in the quality discussion in order to influence the qualities of Information Systems. A framework to an analytical understanding of IS design in an organizational context and of the relations between IS design and quality, is presented.

Keywords
Quality of information systems, participatory design.

INTRODUCTION

In the end, designing quality systems is what Information Systems (IS) design is about. Understanding quality of information systems is, however, a complex matter. Nevertheless, the quality discussion of IS has mainly taken place within the Software Engineering tradition. This has limited the quality effort to concentrate on formal and technical aspects of IS. User participation in system development may also contribute to improved quality, but with a focus on the aspect of software use. Participatory Design approaches IS design in an informal manner (as opposed to more formal ways of Software Engineering).

The two approaches, Participatory Design and Software Engineering, may mutually benefit from each other. The aim of the paper is to bring the two approaches together. To combine ideas from Participatory Design with ideas of the Software Engineering tradition, may create a stimulating and useful synthesis. The two approaches have different methodological repertoires. Whereas Participatory Design is more based in the social sciences, Software Engineering is based in the natural sciences. Combining these approaches is thus a challenge in contents and in method.

In *PDC'96 Proceedings of the Participatory Design Conference.* J. Blomberg, F. Kensing, and E.A. Dykstra-Erickson (Eds.). Cambridge, MA USA, 13-15 November 1996. Computer Professionals for Social Responsibility, P.O. Box 717, Palo Alto CA 94302-0717 USA, cpsr@cpsr.org.

The Participatory Design community may benefit from the Software Engineering tradition with respect to transparency

of decision making, use of documents for contracts, reflection and communication, and use of project management tools and techniques in order to control the design process, bringing Participatory Design into a more realistic frame giving the constraints of Information Systems

Conversely, Software Engineering could benefit from Participatory Design with respect to enabling users to better express and determine requirements and needs of ISs, recognizing the continuous change of IS, thus the necessity of redesign of IS with its methodological challenges (Braa et al, 96). This could make up for some of the major flaws of Software Engineering tradition; taking the requirement specification for granted which may result in developing "wrong" product; and controlling the process by documented routines which may result in bureaucratic (expensive) processes, giving no guarantee of producing a quality product.

The strength of Participatory Design seems to be the weaknesses of Software Engineering and vice versa; Software Engineering with the strength at handling complexity but with the danger of developing a wrong product; Participatory design with the strength of revealing uncertainties of the product but with the danger of not being able to control the process.

However, what both approaches are lacking is the organizational context (and constraints) in which the IS is used. Participatory Design has the main focus on work organization and or selected user groups. Software Engineering focus on software organizations. E.g. Capability maturity Model (CMM) (Humphrey, 90) with the aim of measuring the software organizations ability to deliver software products. An organizational perspective is crucial in order to adapt the IS to the organizations needs and policy. Designing a system that not fit what the organization needing would not be accepted nor useful.

Those of us doing Participatory Design for the last decade, have recognized the problems of getting impact on the final design of Information Systems. Recognizing organizational and technical constraints imply challenges for Participatory Design approaches which need to be addressed.

A framework to an analytical understanding of IS design and of the relations between IS design and quality, is presented (Braa, 95a). This framework for IS design from a quality perspective consists of two parts:

1) An IS Quality framework for understanding and discussing the product quality. The quality framework is a tool for analyzing existing ISs, as well as instances of future products.
2) An IS Design framework for conducting the design process. The IS design framework is an analytical tool for assessing, planning and conducting the design process.

This division is necessary in order to distinguish aspects of the product and process of design. Though the two are closely interrelated, certain aspects should be studied separately.

The quality framework is a means of analyzing existing IS in use in an organizational context, in order to understand the product in an interplay with the work organization. and for analyzing examples of future products, e.g. prototypes (input to the design). The IS design framework is an analytical tool for understanding and conducting the design process, in order to aid the process. Applying the framework is an attempt bringing Participatory Design into a more realistic frame giving the constraints of Information Systems.

PERSPECTIVES ON INFORMATION SYSTEMS QUALITY

IS quality has been on the agenda for long time. There have been a number of attempts to classify quality in purposeful way. Technical aspects of IS quality have been the main focus until recently. (One notable exception is "Computers in Context" by Dahlbom & Mathiassen (93), which takes a philosophical and holistic approach towards IS quality.) However, there has been a slow shift towards the recognition of use quality as an important factor.

IS quality is concerned with how well the software artefacts perform in the organizational setting (Miles, 85). However, organizational perspectives of IS quality are given very little consideration in IS literature on IS quality. And, if considered, it is the software organization which is in focus not the user organizations in which the IS is used. By organizational perspectives of IS, I mean how the Information Systems perform according to the goals, strategy and customer policy of the organization. Within the organizational perspective different interests among user groups have to be coordinated and given different priorities.

Crosby (79) defines quality as "zero defects". This is in line with the popular statement of "doing the right thing first time". This is however not necessarily applicable for systems design; tools and machinery change fast and trying and failing have to be part of the strategy. Within an evolutionary approach to systems design, analysis and design are mutually dependent and performed concurrently (Budde et al, 84; Floyd et al, 89; Kautz, 93), thus "zero defect" will have no meaning. In practise, technical aspects can not be isolated . Another quality "guru" defines quality to be "fitness for purpose" (Juran, 79). In this context, an important consideration is, whose purpose should be satisfied and who is the customer.

Quality of IS depends very much on which point of view it is regarded from. Instead of trying to present a definition of quality, I have characterized IS quality by introducing three important perspectives of quality, each of which represents different traditions for IS. These are Technical quality, Use quality and Organizational quality.

Quality seen from a technical perspective, refers to the software system's structure and performance. Technical quality of a software system is the basis of its functionality (the computer must perform according to expected operations), thus often regarded as the most important. From a technical perspective, quality is "conformance to requirements". The ideal is to make the specification as exact as possible and to make software development a predictable, and thus controllable process. Technical aspects are expressed as measurable properties of a software system, (claimed as being objective), thus easier to specify, than use aspects. From a technical perspective, the need is recognized for measurements that support feedback and control over the development process – "zero defect" is an ideal.

Use quality is seen from a subjective perspective; the end users' actual experience of using an IS. Quality is seen as "fitness for needs" (Kitchenham, 89). From this point of view, use quality and subjective assessments are emphasized in order to evaluate quality. The user satisfaction tradition (Kim, 89), is oriented towards individuals or groups sharing subjective expectations. The ability of performing your work tasks by means of the IS. Within the user satisfaction tradition, high correspondence between expectations and the product signifies good quality, and low correspondence signifies poor quality.

Regarding use quality as an subjective perspective complicates the concept. Different users and user groups will have different experiences of what good quality is.

The intention of focusing on use quality is to provide satisfaction to the end users of the system.

Quality of information systems has another dimension as well. Information systems are operative in an organizational context and social practise, so organizational perspectives on IS quality are also relevant for judging quality. When an information system is well "adapted" to the organization, it can be said to be of high organizational effectiveness, and thereby be of high organizational quality. By "adapted" I mean the match between the organizational strategy (and customer policy) and the output of the information system. The output is a product of the IS (meaning both from software system and work organization) and services (meaning services as perceived from the customer of the organization). Effectiveness is the transformation process performed by the systems meeting the longer term aim (Checkland & Scholes, 90). From the organizational perspective, obtaining effectiveness is the main concern.

Thus, integration of software systems and work organizations is an important issue. Within the organizational perspective different interests among user groups have to be coordinated and prioritized. Identifying the stakeholders is an important issue in order to obtain organizational effectiveness (see section 3.3). The organizational quality perspective adds a new view point, namely how the ISs are functioning in the particular organizations as a whole. Does the IS fit the organizational structure, culture, social practice, goal and customer policy?

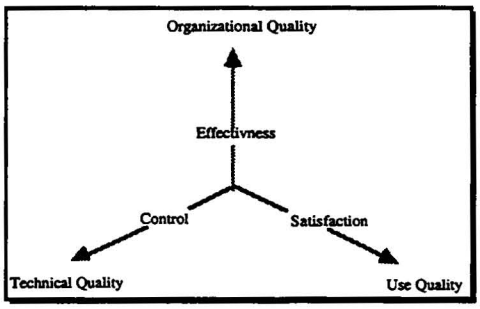

Figure 1: The IS quality Framework for understanding the IS

The framework of information system quality is viewed as x, y and z axis. (Figure 1). The end points represent the previously mentioned perspectives on IS quality: technical quality, use quality and organizational quality. The arrows represent the aim of the perspectives. For example, as the degree of control activities increases, the organizational and use considerations decrease until, at some stage, the perspective appear as solely technical.

In the IS quality framework, these three perspectives on quality are placed as end points in a x, y and z axis illustrating their inter-dependency. When the focus is on technical quality vs. use quality, the organizational quality adds contextual claim as e.g. customer service; when focus is on use quality vs. organizational quality, a certain degree of technical quality is required; when the focus is on organizational quality vs. technical quality, a minimum requirements of use quality is required. The space of the x, y and z axis represent the dilemmas of IS quality. Strategies and techniques for how to address the different quality perspectives in the design will be issued in the design framework.

The Framework Discussed

To illustrate the difference in perspectives when judging quality, let us look at the application of the World Wide Web (WWW) in an educational institution (Braa, 95a). WWW is used in order to provide information services to students, as well as providing registration routines. The technical quality is generally regarded as good: WWW works on different platforms, the "whole world" can be reached, the availability is good and so on. The use quality is not so good; it is difficult to navigate in the system. The information gained easily appears as data overload (information underload), so that it is difficult to find relevant information. However, if we regard the administration personnel as being the users, they experience it as being satisfactory since it reduces the time needed to answer questions from students. When the students seek information or try to register on courses they must do so using WWW themselves. The students are here regarded as customers, since they receive the output from the system, but they might as well be regarded as users. The "customer's" experiences are two-fold: for those who are not used to surfing on the net, the amount of service available has decreased, and these are probably the customers who really need information and support. Others may be happy without direct contact with the staff, in order to receive information. When we analyze the situation, we find a mismatch between the stated policy, which says the institution should support the students so that they find their way through the educational system in a effective way, and the service they experience. The WWW could then be said to not support organizational effectiveness and is therefore plotted close to the origo (Figure 2). The application was designed with close cooperation with those regarded as users. Thus, the perspective of the administration personnel was implemented in the design. If the students had been regarded as users as well, the design of the system could have been different as well as failures could have been avoided, such as the deadlock situation where password is required in order to subscribe to a course, however students do not get a password before you have been confirmed as a student at a course.

The net result lead to so many complaints, that the total number of inquiries from the students is not actually reduced at all. This messiness of identifying stakeholders as users and customers is especially true for Internet applications. Users will represent different perspectives and interests.

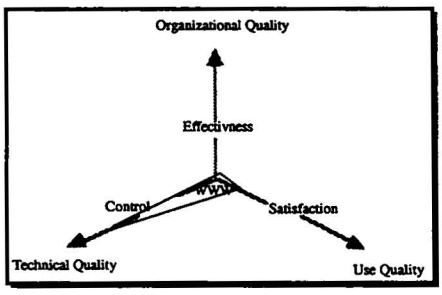

Figure 2: The World Wide Web case plotted in the IS quality framework.

Neither of the perspectives are sufficient when aiming at understand quality of information systems in an organizational context. The relationships between the perspectives are closely intertwined, e.g. a robust technical artefact is needed if the IS is to function effectively and the artefact is used in an organizational setting, for an organizational purpose. If an IS is not used, it does not

have any qualities. How to evaluate use quality will depend from which user (group) it is regarded.

This framework aims at providing an analytical tool for understanding and evaluating the quality of an IS product in an organizational context. Thus, the quality framework is a tool for analyzing existing IS, and for analyzing instances of future products. A focus on products is needed during the system's life cycle, in order to communicate the different perspectives, and to focus on IS in use. Products may thus be prototypes to be used for experimentation, examples of design alternatives for various versions, or simply cognitive constructions. It is important to emphasize the product in order to exchange expectations among the stakeholders, and in order to understand the product as interactive with the work organization. This is particularly important when use quality is to be assessed and communicated. It is also important when the IS product is to be adapted to fit the organization.

Although, product quality is experienced during use, it is during the process of designing information systems that quality is influenced. These two views are closely interrelated and both views should be considered throughout the system's life cycle (See section 4). Making quality and product considerations part of the design process is the issue of the framework presented below.

THE FRAMEWORK FOR INFORMATION SYSTEMS DESIGN (PROCESS)

The IS quality framework constitutes a basis for the IS design framework. Each design perspective relates to a corresponding quality perspective. While the IS Quality framework is abstract and concerned with understanding, communicating and evaluating IS quality, the IS design framework is about carrying out design. In the framework of IS design, the main perspectives to software design are identified and characterized as: Software Engineering, Software Use and Organizational Implementation of IS. In addition, three approaches to IS design are identified (the arrows in Figure 3); Quality Assurance, Participatory Design and Total Quality Management. Each approach supports a different perspectives. The space of the vertexes represent the dilemmas of IS design.

In the following, I will present each perspective and an approach within each perspective to software design. Each approach expresses important considerations but is insufficient by itself.

Software Engineering

Software engineering primarily represents a tradition. Nevertheless, it is regarded here as a perspective on the design process. Software engineering aims to control the process, by use of formal methods and measurements. The ideal is to make the specification as exact as possible and to make software development a predictable, and thus controllable process. This is mainly done by controlling the software development process through documentation routines. From a software engineering point of view, quality is "conformance to requirements" (Crosby, 79; Roetzheim, 88; Sommerville, 89). Software engineering has a primarily technical perspective on software design. This perspective correlates with the production view (Vidgen, Wood-Harper & Wood, 93) where the ideal is the production of a software product that conforms to a specification, has zero defects and is produced as efficient as possible.

The traditional way of conducting design from a Software Engineering perspective is by fragmentation of work into routines, detailed description in order to make the process predictable and controllable, separate management and performance tasks removing skill and knowledge from the workers' area and incorporating it into management functions (cf. Quality Management). These characteristics correlate with characteristics of Taylorism. Part of Taylor's effort was to make work procedures standardized and therefore more predictable (Taylor, 12; Greenbaum, 79).

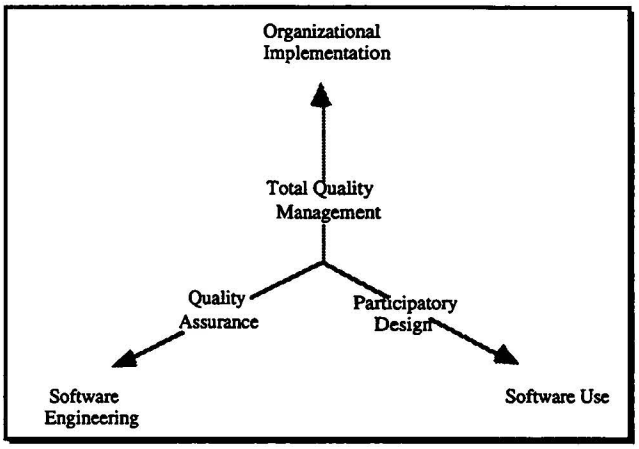

Figure 3: The framework of software design.

Quality Assurance

Quality assurance contributes to making the development process transparent, by allowing people from the outside to evaluate the development activities through audits, reviews and inspections. However, from a use and organizational perspective, these techniques depend on the documents reflecting the actual practice of the system development process, in order to achieve success.

A standard of quality assurance is a specification of a minimum level of quality assurance activities, decided by some standardization body. Standards are generally defined in terms of a model of best practice, against which all other practices may be compared. Quality assurance according to ISO 9001 is an attempt to structure system development by establishing quality systems and quality routines. The ISO 9000 standard for quality assurance (ISO 9001) is here treated as an instance of traditional quality assurance, since the standard builds on the same principles as traditional quality assurance (Braa & Øgrim, 94). The standard is widespread throughout the world. It has been accepted as the harmonized standard since 1992 by all the national standardizing bodies both in the European Union (EU) and

the European standardization organization (CEN). If an organization manages to become certified, the quality activities have to be emphasized over time, due to periodical revisions from the certification body.

ISO's main point is that the process of production should be set up in the same way as it is described. The certification process usually starts with the description of activities related to the system development process. A danger could then be that much effort is spent on the description process, which then implemented as it is, and no time is spent on improvement activities.

ISO 9001 focuses on technical quality, and leaves use quality up to the users' ability to specify (Braa & Øgrim, 94). ISO gives guidelines for "doing things right", but has no emphasis on "doing the right things". ISO 9001 aims to "produce a product according to specification rather than depending on the test and validation activities for assurance of quality". A consequence could be that users will get what they specified, even if it turns out to be of low use and poor organizational quality. An explanation for the ISO 9001's focus on specifications, could be that its origin is the manufacturing of goods, not the production of intangible services such as information systems.

Even though problems with document-driven system development have been reported for some time (Boehm, 88), quality assurance, here represented by ISO 9000, mainly relies on document-driven system development.

The objective of the documentation routines are to document the system to help staff improve the processes through identifying problems, appropriate training, work study etc. With a mindless implementation of the ISO 9000 system, the focus on improvement may be lost and the system become driven by the paper work (Braa & Øgrim, 94). The objective becomes paper work instead of improving the system.

The effectiveness of ISO 9000 as a tool to drive quality is being questioned by the European Commission's Directorate General III for industry, (Zuckerman, 94). They are talking about de-emphasizing certificate programs because they are not helping European firms infuse quality into their organizations.

Software Use

By the term software use perspective, I mean software evaluated from the end-users' point of view when working with the software system. The aim is to increase user satisfaction. Software use is often regarded as an individual matter. This implies that the experience of quality will vary among users and user groups. The perspective is here called software use and not IS use, since emphasis is on the use of the particular software within the information system, such as including text processors, spread sheets and other tools.

The experience of software use is not a static subject. Changes in the work organization will lead to changes in the experience of the system and changes in the software system (even minor) may affect the experience of quality. Consequently, the use aspects must sustain evaluations during the systems' life cycle.

An illustration of the dynamics of software use is the use of text processors. Changes in the level of competence change the experience of what qualities are good in a software system. At first, constant appearance of help dialogues, such as "Do you really want to leave the system?" feel secure and informative. But after a while, one gets annoyed and distracted by such interruptions and regard them as noise.

From the perspective of software use, high correspondence between expectations and experienced software use signifies good quality, and low correspondence signifies poor quality. Users participating in design often create high expectations among the users, both with respect to use quality and delivery time (Floyd, 84). However, the "expectation view" gives the developers the opportunity to manipulate the users. Original expectation can be decreased under the cover of technical limitations. The degree of user satisfaction can be high, even if the system is of low technical quality. Low technical quality reduces the possibility to maintain and change the system. When user requirements change, these changes may therefore be impossible to implement. This means that the quality of use experienced decreases after a period of time.

Participatory Design

Participatory Design (PD) is an approach which supports a software use perspective. PD represents an approach towards computer systems design in which the people destined to use the system play a critical role in designing it (Schuler & Namioka, 93). In the Participatory Design approach, we find related "approaches" such as user participation (Ehn, 88; Mumford & Weir, 79), prototyping (Floyd, 84), cooperative prototyping (Grønbæk, 91) and cooperative design (Greenbaum & Kyng, 93). The approaches do not represent distinct differences for this purpose, even though user participation also cover users as representatives and information providers and does not necessarily focus on actual participation in designing solutions. PD is connected with a set of participatory techniques but also a set of values concerning human aspects such as the relationship between technology and human activities that provide technological systems with their reason for existing.

There are several reasons why users should participate in system development. The reasons can be roughly grouped into two: 1) because it is a democratic right in Scandinavia (those who are affected by the technology should affect it) and 2) computer systems will be better in use (fit the work tasks better) if users participate in the design. However, giving the constraints of IS design Participatory Design has often little impact. Well aware of the advantages of doing Participatory Design I have experienced several problems which may cause major problems (Braa, 92; Braa, 95a; Braa, 95b).

- Lack of control over the design process resulting in much effort spent on arbitrary comments from users.

- Lack of control over the decision making process due to little adaptation of traditional management techniques and documentation.
- Lack of documentation; When decisions are not documented, it is difficult for others to evaluate them and it could easily become arbitrary. Documents are important when it is necessary to mediate and maintain a common comprehension between team members coming from different organizational cultures, as is often the case with users and developers.
- This may result in lack of transparency for those not participating.
- Lack of recognition of technical aspects of the IS.
- Lack of recognition of organizational constraints and policy.

The focus in participatory design has mainly been on system-design project groups (Braa, 95a). This view can be too narrow, excluding organizational interests such as co-ordination activities and more overall goals. Users who are participating in the projects can typically influence the design to a limited extent. However, this is not sufficient if the development takes place in a context of multiple clients or multiple users. Different user groups may have quite different views on what good quality is. In such a situation, software use perspective becomes more of an organizational matter, and an organizational perspective is necessary.

Organizational Implementation

When a software system is well adapted to the organization, it can be said to be of high organizational effectiveness, and thereby be of high organizational quality. The focus is on effectiveness rather than efficiency. A transformation process which works and uses minimum resources might still be regarded as unsuccessful if it were not achieving the longer term aim. The effectiveness of the information system on the organizational level is an important issue in the design of quality information systems.

It is well known that organizations invest a lot of resources in information technology. It has been difficult to document and measure profits, which are not due to rationalization and increases in efficiency, such as service quality and work satisfaction. The focus has then often been on rationalization profit in order to justify the investments. The relationship between information technology investment and what the organizations expect to gain from it should undergo thorough analysis.

In order to obtain high organizational quality, it is important to reveal who the customers are and who the stakeholders are. Stakeholders of an IS are the "victims" or "beneficiaries" of the information system. The concept of stakeholder encompasses customers, but because it is wider in scope, there is less risk of missing something significant (Vidgen et al, 94). There will be many stakeholders involved, and it is not necessarily the most important stakeholder who should play the role as the customer. Vidgen et al (93) suggests stakeholder analysis (Mitrof & Linstone, 93) to make the assumptions of different stakeholders explicit. Vidgen (94) combines stakeholder analysis with soft systems thinking (Checkland, 81; Checkland & Scholes, 90) in order to gain an understanding of what quality means to them. Different stakeholders may have different interests, which need to be coordinated, negotiated and given different priorities.

Total Quality Management

One way of obtaining organizational quality is by Total Quality Management. TQM can be said to be the only approach which is mainly concerned with an organizational implementation perspective. TQM is an approach to improve the effectiveness and flexibility of an organization as a whole (Oakland, 89: Levit, 94). As its name indicates, TQM aspires to build quality into every conceivable aspect of organizational activities. However, in reality, the theory and practise of TQM continue to be dominated and constrained by the orientations in the disciplines and the preoccupation's of its major advocates (e.g. Deming, 86; Juran, 79; Feigenbaum, 83). With backgrounds in operational research and statistical methods of control, many of the leading "gurus" sought to develop and refine objective means, such as statistics in order to gain "hard" information about production processes and service delivery. Much attention and effort has been directed to the measurement and documentation of procedures and outcomes (Wilkinson & Willmott, 95). Comparatively less consideration is given to the "softer" process of winning support for and commitment to the TQM philosophy of continuous improvement (Wilkinson et al, 92). This has begun to shift with the new generation of TQM advocates. Oakland (89) stresses the importance of getting the employees to become committed to attaining quality in a highly motivated fashion.

TQM introduces a focus on the customer as an explicit stakeholder when it comes to the quality of information systems, as an indirect consumer of the output from the information system. In the end, it is the customers who pays for low IS quality. If, for instance, an air plane ticket system is of low use quality, the customer will have to wait a long time for the expected service, and if the technical quality is also poor in terms of availability, the ticket may not be booked on time. The customer may choose another air line, or pay for low quality through overpriced tickets.

TQM is fairly consensus oriented: different interests and possible conflicts of interest are not dealt with. When assessing quality of information systems, questions of economy, power and interests will sooner-or-later arise: are the computer systems developed for the interests of individual users, for groups of users or for the organization as a whole? The importance of regarding different interests is discussed in the literature (Briefs et al., 83; Bjerknes et al., 87). Different user groups may have diverging, maybe contradictory interests of what signifies good quality. A personnel control and wage system might represent good quality of use for management and the personnel department, but not for the controlled workers. The result

could be high turnover, and thus influence organizational quality. The different groups of users make different demands on functionality, in order to support their work. If these different interests are not met, but rather an intersection of several interests is catered for, the results could be a computer system which is not suited to anyone and will be of low efficiency for the organisation as a whole.

One of the main problems with TQM is that it is primarily a philosophy, not a method or a set of techniques, thus difficult to apply. TQM is very much aimed at management level (therefore easy to sell). There has been little analysis of its results.

THE FRAMEWORK FOR INFORMATION SYSTEMS DESIGN FROM A QUALITY PERSPECTIVE

Two different parts of the framework have been presented, each of which describes different aspects of IS. In the IS quality framework, three perspectives on quality are identified. It is argued that neither of the perspectives are sufficient when aiming at creating a holistic picture of quality of information systems in an organizational context. Understanding quality of IS is connected with being able to visualize ISs as products in one way or another. Communication and learning about quality is mediated through intermediate products such as mock-ups, prototypes or documents of various kinds (such as specifications). Knowledge and understanding of IS quality evolve through the development process. Thus, the IS quality should be emphasized throughout the design process. This can be done through evaluations and revealing expectations by means of the IS quality framework.

Three perspectives on the design process are identified in the IS design framework: Software Engineering, Software Use and Software Implementation. Applying all three perspectives are recommended in order to cover the complexity of an in-context design process giving the constraints of IS design. If the design process is positioned biased towards one perspective, this must be an intentional, conscious decision, not by accident (knowing that important aspects will be lacking).

The parts of the framework interrelate in so far as each perspective and arrow correlates with the other framework respectively; the Software Engineering perspective holds a technical view on quality, control of the process is the main strategy of quality assurance, the software use perspective supports a use perspective, Participatory Design aims at user satisfaction, and Organizational Implementation supports an organizational perspective on quality.

Software engineering tends to suggest that quality may be achieved solely as a product of good engineering practice. Total Quality Management emphasizes the importance of people, culture and process management. It does not, however, take into account the peculiar nature of software. Software is unusual because it does not have a physical existence in the same way as manufactured artefacts. Participatory Design approaches often fail to tailor the design to one user group omitting organizational and technical constraints and needs.

An example of this phenomenon is the case of the highly participatory project where all the end-users were represented in the design project and collaborated in the design of a book-purchasing system for a library (Reijonen, 95). The result was a system of very low quality with respect to organizational and technical quality. However, the end-users were very satisfied with the system in use. It turned out that the users when designing the system were particularly concerned about maintaining the existing work organization i.e. the division of labour. Because of this, other maybe more optimal solutions were neglected. Customer service, which is a prerequisite for obtaining organizational effectiveness in a library, was not a main concern of the users' daily work, and were therefore given little attention.

In Figure 4, the inter-relationship between these parts is shown and presented as a framework for IS design from a quality perspective. IS as products are not viewed as end products (never finished) but exist in terms of their use, which again create new instances of products (e.g. versions, redesign).

Placing the two parts together focuses on the necessity of applying both a process view and a product view in IS design. A certain comprehension of the product is especially essential conversely it is difficult to determine and assess the expected quality of the product. This is true especially if there is high uncertainty as to how the product should be. Without a process view the complexity of the variety of software use and organizational implementation will be lacking. Prototyping serves two functions: prototyping as a learning process and prototypes as products to be evaluated. Experiments with prototypes both in a lab situation and within the real working domain are important, when it comes to use and organizational concerns. It is, however, important to emphazise the product as well, in order to gain a common comprehension among the stakeholders, and in order to understand the product in an interplay with the work organization. This is particularly important when use quality is to be assessed and communicated. It is also important when the IS product is to be adapted into the organization. A common assumption is that a good process leads to a good product. This is a reasonable assumption when the product is well defined, understood and described, as in goods production. In IS design, however, the product of each development process is unique. It cannot be fully specified in advance. If the quality information system is not understood, it is difficult to design an information system of high quality. The presence or absence of quality is experienced through use of a product not in the process of producing it.

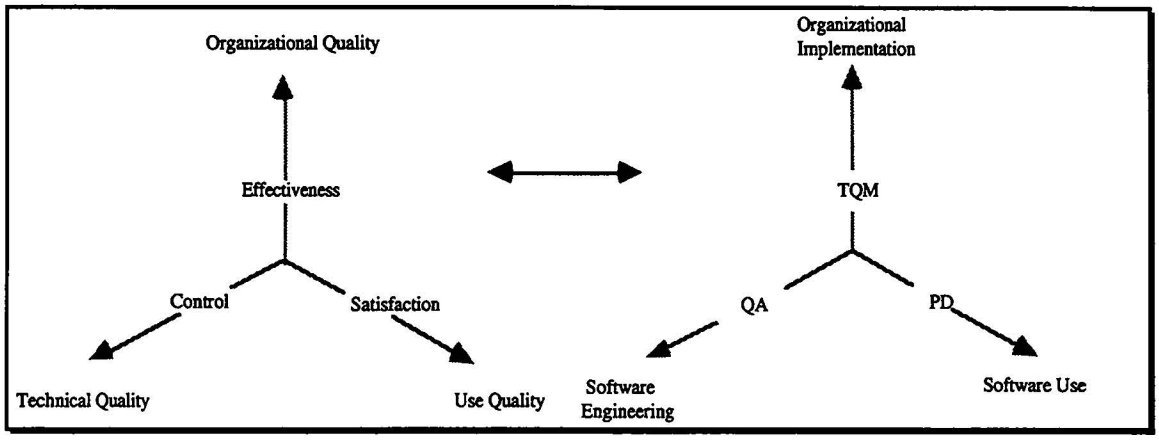

Figure 4: The framework for IS design in a quality perspective.

It is, however, important to document the results in such a way that ensures they will be implemented e.g. requirement specifications. There is a continuously dynamic relation between the product instances and the processes of designing, using and implementing the IS in an organization.

In the context of IS development and IS redesign the aim is to secure that all three perspectives are attended to in a conscious way. Although, product quality is experienced through using the product, quality is influenced during the process of designing information systems. These two views are closely interrelated and both views should sustain consideration throughout the system's life cycle. Making quality and product considerations part of the design process is the aim of the framework.

The intention has been:

- To move emphasis towards consideration of organizational implementation away from a biased software engineering and a biased software use perspective.
- To supplement the repertoire of existing IS quality techniques to include experimental techniques for IS design.
- To complement the participatory design tradition with ideas of transparency, control and documentation work from the software engineering tradition.

Acknowledging and regarding the constraint (technical, economical, organizational) in which limit and make Information Systems possible, is necessary for Participatory Design approaches in order to influence the final product.

The context and constraints for IS development are changing fast, and experimental techniques are flexible to adapt to new situations both in terms of technology, use situations and organizations. Future research is needed in order to provide IS design with additional experimental techniques for creating usable systems of high quality – quality that reaches beyond what can be gained with formal methods.

REFERENCES

Bjerknes, G., Ehn, P. & Kyng, M. (1987) Computers and Democracy: A Scandinavian Challenge, Avebury Gower Publishing Company Ltd, Aldershot.

Boehm, B. W. (1988) A Spiral Model of Software Development and Enhancement, in IEEE Computer, May, pp 61-72.

Braa, K. (1992) Influencing System Quality by Using Decision Diaries in Prototyping Projects, in Muller et al. (eds) Proceedings of the Participatory Design Conference, pp 163-170, Cambridge MA US.

Braa, K. (1995a) Beyond Formal Quality of Information Systems Design, Dr. Scient Thesis, June 1995, Department of Informatics, University of Oslo.

Braa, K. & Øgrim, L. (1994) Critical View of the ISO Standard for Quality Assurance, in Information Systems Journal., Vol 5, pp 253-269 .

Braa, K., Bratteteig, T., Øgrim, L. (96) Organising the Redesign Process in System Development, to appear in Special Issue on Information System Development of Journal of Systems and Software, May, 1996.

Briefs, U., Ciborra, C. & Schneider, I. (1983) Systems Design For, With and By the Users, North-Holland Publishing Company, Amsterdam.

Budde et al (eds)Approaches to prototyping, Springer-Verlag, Berlin.

Checkland, P. & Scholes, J. (1990) Soft Systems Methodology in Action, John Wiley & Sons, New York.

Checkland, P. (1981) Systems Thinking, Systems Practice, John Wiley & Sons, Chichester, New York, Brisbane, Toronto.

Crosby, P.B. (1979) Quality is Free, McGraw-Hill, New York.

Dahlbom, B. & Mathiassen, L. (1993) Computers in Context. The Philosophy and Practice of Systems Design, NCC Blackwell, Cambridge, MA & Oxford, UK.

Deming, W.C. (1986) Out of the Crisis, Cambridge, MA, MIT Center for Advanced Engineering Study.

Ehn, P. (1988) Work-Oriented Design of Computer Artefacts, PhD Theses, Arbetslivscentrum, Stockholm.

Feigenbaum, A.V. (1983) Total Quality Control New York, McGraw-Hill.

Floyd, C. (1987) Outline of a Paradigm Change in Software Engineering, in Bjerknes et al (eds)Computers and Democracy, pp 191-210, Gower Publishing Company, Vermont.

Floyd, C., Reisin, F.-M. & Schmidt, G. (1989) STEPS to Software Development with Users, in ESEC 1989, pp 48-64, Springer Verlag, Berlin.

Gillies, A. C. (1992) Software Quality. Theory and Management, Chapman & Hall Computing, London.

Greenbaum, J. (1979) In the Name of Efficency - Management Theory and Shopfloor Practice in Data-Processing Work , Temple University Press, Philadelfia.

Greenbaum, J. & Kyng, M. (eds.) (1991) Design at Work: Cooperative Design of Computer Systems, Lawrence Erlbaum Ass, New Jersey.

Grudin, J. (1989) Why groupware applications fail: problems in design and evaluation, in Office, Technology and People, vol 4 no 3.

Grudin, J. (1994) Computer Supported Cooperative Work: History and Focus, in IEEE, January.

Grønbæk, K. (1991), Prototyping and Active User Involvement in System Development: Towards a Cooperative Prototyping Approach, Ph. D Thesis, Computer Science Dept., Aarhus University.

Hammer, M. & Champhy, J. (1993) Reenginering the Corporation, A Manifesto for Busniness Revolution, Nicholas Brealey, London.

Humphrey, W. S. (1990) Managing the Software Process, Addison Wesley, Reading, USA.

Jacobsen, E., Lien, G.I., Melling, O.A., Tangen, K.L. (1991) Computer Use in the Norwegian Institute of Town and Regional Planning (in Norwegian) Department of Informatics, University of Oslo.

Juran, J.M. (1979) Quality Control Handbook McGraw-Hill Book Company, New York.

Kautz, K. (1993) Evolutionary System Development. Supporting the Process, Research Report 178, Dr. Philos Thesis, University of Oslo, Department of Informatics.

Kim, K. K. (1989) User Satisfaction: A Synthesis of Three Different Perspectives, in Journal of Information Systems.

Kitchenham (1989) Software quality assurance, in Microprocessors and Microcomputers, Vol. 13 (6), pp 373-381.

Levit, S. (1994) Quality Is Just the Beginning – Managing for Total Responsiveness, McGraw-Hill, New York.

Miles, R. (1985) Computer Systems Analysis:the constraint of the hard systems paradigm in Journal of Applied Systems Analysis, vol. 12.

Mitrof, I. & Linstone, H. (1993) The Unbounded Mind: new thinking for the 21st centery, Oxford University Press, New York.

Mumford, E. & Weir, M. (1979) Computer Systems in Work Design - The ETHICS Method, Ass Buisness Press.

Oakland, J. S. (1989) Total Quality Management, Butterworth-Heinemann, Oxford.

Reijonen, P. (1995) Moderately satisfying but hardly effective, in Dahlbom et al. (eds) Preceedings of the 17th IRIS, Aarhus, Denmark.

Roetzheim, W. H. (1988) Structured Computer Project Management, Prentice Hall, Englewood Cliffs, New Jersey.

Rothery, B. (1992) ISO-9000, Gower Publishing Company.

Schuler, D. & Namioka, A. (ed.) (1993) Participatory Design Principles and Practices, Lawrence Erlbaum Ass, New Jersey, US.

Sommerville, I. (1989) Software Engineering, Addison.Wesley Publishing Company, Wokingham, England.

Taylor (1912) Scientific Management in Pugh, D. S. (1971) Organization Theory selected readings, Penguine Books.

Vidgen, R. T., Wood-Harper, A. T. & Wood, J. R. G. (1993) a Soft System Approach to Information Systems Quality in Scandinavian Journal of Information Systems, Vol 5, pp. 97-112.

Vidgen, R., (1994) Information Systems: identifying customers and understanding quality requirements, in Precedings of the 17th International systems Research seminar In Scandinavia - IRIS.'17, Finland.

Vidgen, R., Ross, J. Wood, B., Wood-Harper, T. (1994) Buisness process reengineering: the need for a methodology to revision the organization in TC8AUS

IFIP information Systems International Working Conference, Gold Cost Queensland Australia.

Wilkinson, A & Willmott, H. (eds.) (1995) Making Quality critical New Perspectives on Organizational Change Routledge, New York.

Wilkinson, A., Marchington, M., Goodman, J. & Ackers, P. (1992) Total Quality Mangement and Employee Involvement, Human Resousce Management Journal, Vol 2 no. 4, pp 1-20.

Zuckerman, A. (1994) EC drops ticking bomb: IT could prove lethal to the ISO community in Industry Week Magazine and Quality Progress magazine.