Quality IS: Next Most Important Research Project?

Birgitta Fagerström, Ulrica Löfstedt & Ia Williamsson
Linnaeus University, Växjö and Mid Sweden University, Östersund
birgitta.fagerstrom@lnu.se, ulrica.lofstedt@miun.se, ia.williamsson@lnu.se

Abstract. High quality information and updated information systems are essential for professional activities. Software maintenance plays an important role in keeping a desired level of quality and during recent decades there have been different efforts done in this area. The purpose of the study presented in this article, is to find relevant research questions and approaches for further research. A theoretically focused study of a limited number of articles in the areas of software maintenance, systems maintenance and IT-governance was conducted. The direction of the research presented in the articles indicates that development of new models and approaches is of great interest for researchers, but empirical studies have had low priority. The common experience of the study can be divided into four areas of concern: models, motivation and validation, definition uncertainty, and use of concepts from research. It is hard to make any conclusions concerning quality IS. However, it is obvious that the area is of interest for research and some suggestions for further research were identified.

Keywords: Quality IS: IS maintenance – software maintenance – management and IT-governance, models and frameworks, intended user

1 Introduction

Information quality (IQ) and useful information systems (IS) are essential for both individual-, organizational- and social performance. We have reached a point where we demand and are dependent on good quality information from current and relevant IS. The area of software maintenance is closely connected to this demand and research has over the years produced a range of models and frameworks in order to guide practice in utilizing the information assets (e.g. DeLone and McLean, 1992; Kajko-Mattsson, 1998; Kitchenham, 1999). It is however hard to grasp the overall picture of existing knowledge since research has approached the area of software maintenance from different angles and areas of knowledge and consequently solved the same kinds of problems in different ways using different definitions and concepts.

The original purpose of this project was to describe and analyze definitions on quality IS. Indicators from the areas of software maintenance, systems maintenance and IT-governance
was searched for in order to establish a baseline for two different research projects (in Swedish: “mjukvaruunderhåll”, “systemförvaltning” and “IT-styrning”) This attempt ended when we realized that the large amount of definitions, frameworks and concepts was impossible to analyze and structure the way we intended. The purpose of current project is to analyze a limited number of articles in the areas of software maintenance, systems maintenance and IT-governance in order to find relevant research questions and approaches for further research.

2 Method

A semistructured article review was conducted. One common keyword, information quality, was used in combination with our areas of interest: software maintenance, systems maintenance and IT-governance. Our workprocess proceeded as follows: Four articles each were selected based on individual preferences. The articles were documented in a common template with aspects listed in Table 1.

<table>
<thead>
<tr>
<th>Aspect</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purpose</td>
<td>Explicit or implicit</td>
</tr>
<tr>
<td>Research questions</td>
<td></td>
</tr>
<tr>
<td>Method</td>
<td>Kind of study - casestudy/small n-study, survey other</td>
</tr>
<tr>
<td>Empirical ground/base</td>
<td>How extensive empirical data? - What kind of empirical data?</td>
</tr>
<tr>
<td>Definitions</td>
<td>Describe definitions related to system quality?</td>
</tr>
<tr>
<td>Concept for systems quality</td>
<td>What is included in this concept according to the article?</td>
</tr>
<tr>
<td>Result and Conclusions</td>
<td></td>
</tr>
<tr>
<td>Any discussion about practical use/practical application of concept/s in the article?</td>
<td></td>
</tr>
<tr>
<td>Other</td>
<td></td>
</tr>
<tr>
<td>Reflections</td>
<td></td>
</tr>
<tr>
<td>The thoughts / ideas raised by this article</td>
<td></td>
</tr>
<tr>
<td>Suggested further research</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Common template with focus aspects for article reading

After this an analyze phase followed where comparisons and discussions on our findings took place. A process of open discussions resulted finally in a common experience documented as discussion and reflections. This kind of comprehensive study on a specific topic can according
to May (2002) be referred to as a theoretically focused study that chooses a “carefully targeted sample” that is well positioned to clarify the issues under analysis. The sampling strategy provides an efficient way to answer broad questions with a comparatively small set of article/literature choices.

3 Related literature

This section introduces the discussion on definitions of quality IS, software maintenance and IT-governance. Since different kinds of models play an important part in almost all articles a paragraph on how to interpretate models is included.

3.1 Definitions

In the section below some definitions concerning quality, software maintenance, systems maintenance, and IT Governance are presented.

There are a lot of factors connected to the concept of quality. According to CisQ (Consortium for IT Software Quality) (n.d), five factors are needed to provide quality. These factors are reliability, efficiency, security, maintainability and size. Ohlsson (1997) has made a summary of quality factors and according to this the most frequent concepts present in different definitions are: Correctness, reliability, efficiency, usability, integrity, maintainability, flexibility, testability, portability, resusability, and interoperability. Accordingly, quality of software can be defined in different ways. In the ISO standard for software engineering, usability is defined as; the capability of the software product to be understood, learned, used, and attractive to the user, when used under specified conditions (ISO 9126-1:2001, 6.3). A user is an end user as well as an indirect user (dependent on the software), and usability can be affected by functionality, reliability and efficiency.

The software maintenance standard ISO/IEC 14764:2006, describes software maintenance as “the totality of activities required to provide cost-effective support to a software system”. This definition is referred to as software management (systemförvaltning) by Brandt (2005). He also state that software maintenance might apply on software (the program) as well as on (the information) system. The SS-ISO/IEC 90003:2005 definition of software maintenance is; “software product undergoes modification to code and associated documentation due to a problem or the need for improvement. The objective is to modify existing software product while reserving its integrity”. The maintenance activities performed can be described as corrections and as enhancements. Corrections are the actions of keeping the software in an agreed quality level, and enhancements is about enhancing functionality and maintainability of the software, April (2004).

The swedish concept “systemförvaltning” seems to be equivalent to the english concept “software maintenance”. It has been used since the 1970’s and even though almost 50 years have passed, the meaning is almost the same. The meaning of “system” in the swedish version and “software” in the english version is not explicitly defined in literature. The definitions of limits or scope of “software maintenance” might be; program, software or system, and the
“maintenance” might refer to maintenance, administration or management (Brandt, 2005). Further he recommends to use the definition of “software” in its immediate context. An interesting synthesis of definitions of “systemförvaltning” in Brandt (2005) reveals that some researchers tend to identify the concept in a corresponding english term, while others tend to define systems maintenance, software maintenance, and software maintenance management.

According to Weill & Ross (2004) is IT Governance a concern on decision rights for IT-investments. Content in decisions is a responsibility for management. According to Pearson & Saunders (2010) IT Governance is about distributing decision rights in different ways to assist different decision making approaches (i.e. centralized, decentralized, or hybrid). IT Governance Institute uses a broader definition in that IT Governance should make sure that IT in the organisation supports and extends the strategies and objectives of the business. It also addresses the responsibility of the board of directors and executive management, and emphasize the constitution of IT Governance as being leadership and organizational structures, as well as processes (ITGI, 2012).

3.1 Kinds of models

The information systems area produces, uses and have preferences for different kinds of models. Model discussions and definitions are consequently a common feature in articles on systems development. IT-management oriented articles exhibit the same kind of fondness for models but definitions and discussions are rare. The impression is that the models are taken for granted. Brier (2006) outlines a taxonomy of different kinds of theories (or criterias for) what is regarded as truth in different kinds of models. This taxonomy might be useful when discussing models and quality IS. In this taxonomy correspondence theory argues that a theory is truth when it in a model form or in a description match or is consistent with the real world phenomenon. The pragmatic criteria states that a theory/model/mapping is truth if works satisfactory in relation to its purpose. Consensus theory states that the result an infinite group of people will reach if they work with the problem forever is truth. The problem is thou that no result can reach this criteria form the beginning. Coherence theory or context theory note that a truth scientific theory must – as contradiction-free as possible - be linked to to other theories in the field. A sign on a truth theory is that it is linking to other theories and is part of a “theory community”, that the theory in question is a part of a whole. Evidens-(truth) theory argues finally that what is true will reveal itself though evidence (Brier, 2006). 

4 Article content

The twelve on individual preferences selected articles are summarized in this section. Table 2 shows the articles sorted by focus of interest. Each article will then be presented using some of the aspects in the common template presented in Table 1. We have used the article search order for the presentation even though the categories “software maintenance” and “systems maintenance” partly covers the same topic due to definition confusion.
4.1 Software maintenance


**Motivator:** More detailed types of maintenance activity naming are needed for researchers and practitioners for using an appropriate naming with a common meaning.

**Problem:** Practitioners use broadly inclusive terms such as ‘perfective maintenance’ in comparison to skills and situations and practitioners report a need to reveal better accuracy in defining correct maintenance activity for researchers as well as for practitioners. In business there are more detailed types of maintenance activities described than in research. It is useful
to have it more detailed when e.g. cost allocation, performing quality assurance, supervising analysts and programmers, planning staff etc.

**Empirical base:** 55 journals and conferences and standards.

**Result:** A sort of model or flowchart/algorithm for identifying types of maintenance by asking questions.

**Practical application:** Building a sort of model or flowchart/algorithm for identifying types of maintenance by asking questions.

**Suggestions for further research:** No suggestions.


**Motivator:** Critical systems are getting more complex and omnipresent, and need more and more maintenance due to high quality requirements

**Problem:** A software maintenance metric (as Defect Fixation Efficiency metric, DFE) is introduced and discussed. This metric measures the efficiency of defect fixation process, i.e. to fix the defect effectively. It will enhance the chance of injection of new defects, and reduce no of delinquent defects. A matrix is also developed to determine the relationship of defect fixation efficiency with the delinquent fixes, customer satisfaction and maintenance phase. It is used as an indicator of the quality of the software.

**Empirical base:** The study uses an example from an IBM development project (defects found, removed, and rejected) to test the new metric.

**Result:** A new software maintenance metric (DFE) that measures the efficiency of defect fixation process

**Practical application:** The metric is tested in a small case and reveals a practical relevans and quality trend in that; the higher DFE the better effectiveness of maintenance and development, and fewer defects in next phase.

**Suggestions for further research:** The authors state a desire to test the metric on a live project in all phases of software development process (from requirement to maintenance). A software prototype to automate defect fixation process will also be target for further research.


**Motivator:** Previous software maintenance processes are “informal or too focused on a specific goal” and dynamic aspects need to be added.

**Problem:** Previous software maintenance processes are “informal or too focused on a specific goal”, therefore an abstract (semi-informal) ontology for software maintenance that extends Kitchenhams’ more static ontology with dynamic aspects (workflows) is developed. Authors state that (in comparison with Kitchenham’s ontology); “our ontology pretends to be more practical and oriented towards managing maintenance projects from a point of view of a business process.”

**Empirical base:** Authors are using action-research for validating the ontology. The ontology developed here is used in three different projects, eg. one in a company dealing with software maintenance outsourcing.

**Result:** Six diagrams (classes) are developed that describes the ontologies and subontologies in detail. The ontology developed has been used in projects and therefore it has a practical
character. This ontology doesn’t only focus on static features and a particular problem, but also on dynamic features and SMP.

**Practical application:** Authors state that “our ontology pretends to be more practical and oriented towards managing maintenance projects from a point of view of a business process.”

**Suggestions for further research:** The authors will focus on the stages of planning and implementation of the ontology in future research. This article only described the design of the ontology.


**Motivator:** A desire to gain better understanding of the nature of information systems quality and to define how it differs from software quality.

**Problem:** “The desire to be able to distinguish software quality from information systems quality triggered the exploratory research discussed in this article”. It is more of an investigation to gain better understanding of the features of information systems quality, and to describe how it is different from software quality. Software quality is about technical characteristics (and commonly the maintenance of systems quality is under-developed), but software performance is judged by users and that is not a technical aspect.

**Empirical base:** Three researchers: one professor in information systems (organisational viewpoint), one professor in information technology management (managerial viewpoint), and one professor in computer science (software engineering viewpoint). Journal articles and standards: 19.

**Result:** The main result is that the borderline between ISQ and SWQ is not clear and the three viewpoints (managerial, organisational, and engineering) must cooperate, they all address IT utilisation in business, but emphasising it differently, using different tools.

**Practical application:** There is no direct practical use of the result since it is a discussion from interview results, concerning differences and similarities of ISQ and SWQ.

**Suggestions for further research:** No suggestions.

4.2 Systems maintenance


**Motivator:** The systems for software maintenance developed are not useful for in-depth analysis and monitoring of software problems

**Problem:** Organisations have to further develop their systems for software maintenance.

**Empirical base:** Three stages: (1) “establish the overall properties of the system studied”, (2) synthesizing of the local models into a “general local model for the concept at hand”, (3) “the synthesized local model were integrated into a general global conceptual model of SM”.

**Result:** A general conceptual model of software maintenance. The foundation of the model is a comparative study of four maintenance models: the SEI quality framework and three maintenance models from three different organisations. The aim of the model is to “provide some guidance to organisations building or improving software maintenance systems”, and to enable better communication and discussions concerning software maintenance.
**Practical application:** The intention was that their conceptual model “will assist in formulating future maintenance and quality frameworks”. The model “can assist in communication and in reasoning about software maintenance.” “It can provide some guidance to organizations building or improving software maintenance systems.”

**Suggestions for further research:** Test and further developement of the model.


**Motivator:** There are a lack of process models in the area of software maintenance

**Problem:** The few existing process models are too general for covering all aspects in software maintenance.

**Empirical base:** Empirical studies of industrial corrective maintenance processes in one large organisation.

**Result:** Presentation and motivation of a process model for corrective software maintenance: CM³ (Corrective Maintenance Maturity Model). The aim of the model is to provide visibility to the process of software maintenance, and to “(1) to establish a common forum for communicating about corrective maintenance, (2) to enable industrial organization’s examine their own practices and compare notes, (3) to provide guidance to industrial organizations in the process of building or improving their corrective maintenance processes, and (4) to provide a pedagogical tool for universities and industrial organizations in the process of educating their students and software engineers within the area of corrective maintenance.”

**Practical application:** The expectations is that CM³ will be an integral part of an model covering all categories of maintenance, a “Maintenance Maturity Model (M3).”

**Suggestions for further research:** None.


**Motivator:** There are a lot of different views of what ‘maintenance’ is about and to be able to discuss the area, there was a need for an ontology of maintenance, covering different opinions and experiences concerning maintenance. The ontology would be a hierarchy of terms as well as a framework for the domain concerning the factors affecting maintenance. It would also offer taxonomy.

**Problem:** The many different views of ‘maintenance’.

**Empirical base:** None.

**Result:** A number of domain factors that they believed influence the maintenance process were identified: Maintenance activity types, peopleware, product, and process organization. This was the starting point for the ontology developed.

The maintenance ontology:

- A product ontology: Product, product upgrade and artifact.
- Maintenance activities: activity, investigation activity, modification activity, management activity, quality assurance activity, and resource.
- Software maintenance process: Two processes
- Peopleware

**Practical application:** The ontology should be a tool for researchers, especially those conducting empirical studies in the area. The aim of the ontology is to assist researchers to “report sufficient contextual detail for other researchers and practitioners to understand the results of empirical studies.” The ontology should “provides useful insights into the type of information researchers should report if we are to understand fully the results of empirical studies of maintenance”.
Suggestions for further research: None.


Motivator: A need to consider software maintenance when including improvements of e.g. quality in business objectives.

Problem: Software maintenance is an expensive and often under estimated part of a systems life-cycle.

Empirical base: Data collection by a large number of questionnaires from “both users and maintainers of software systems, as well as from I/S managers and the user managers and the user managers supported by the software.”

Result: ”Reuse within the application contributes to overall quality of the application being maintained.” “Reuse from other applications and independence of data elements have a positive impact on overall quality.” “Coordination between users and maintainers positively contributes to the quality of change, while the expertise of maintainers and users negatively relates to quality of maintenance changes.”“…structured design, independence and consistency of data elements, and design for change during the development process are positively related to software maintenance quality.”…Indications that “the existing findings on software quality during the development process can legitimately be extended to ongoing maintenance.”…”Maintainer-user coordination is key to achieving quality goals during maintenance.”…”Techniques used in both software development and software maintenance significantly impact quality during the maintenance phase of the life-cycle. The relationship and coordination between maintainers and users is also a key quality factor.”

Practical application: “Three different perceptual indicators of quality were used to assess I/S team performance; quality of application maintenance, overall application quality, and quality of maintenance changes.”

Suggestions for further research: None.

4.3 IT Governance


Motivator: IS assessment is not well established and more research is needed

Problem: Lack of a comprehensive IS success model. This paper examines the need for IS assessment and suggests a comprehensive IS assessment framework linked to organizational performance using existing IS assessment theory as a base and incorporating measurement concepts from other disciplines

Empirical base: Six pages of references, two columns per page

Result: Progress in the development of a comprehensive framework for IS assessment. A contingency model “IS Assessment Selection Model” is developed where service quality is included.

Practical application: Pracitcal application is a driving force. The question on what kind of measures would be convenient for managers pervades the work

Suggestions for further research: Additional research is suggested to advance the IS assessment contingency theory.

**Motivator:** The myriad of variables need to be reduced

**Problem:** "Different researcher have addressed different aspects of success, making comparisons difficult and the prospect of building a cumulative tradition for I/S research similarly elusive”…(p. 60)

**Empirical base:** A literature study using 180 references. The articles cover the period 1981-1987.

**Result:** Categories of IS success defined: System Quality, Information Quality, Information Use, User Satisfaction, Individual Impact and Organizational Impact. There is no consensus of the measures of IS success. Not enough MIS field study research attempts to measure the influence of the MIS effort on organizational performance. MIS is a multidimensional construct and it should be measured as such. The I/S success Model (The “De Lone and Mclean model”) is a result. The taxonomy presented should be useful in guiding future research efforts.

**Practical application:** Yes, this is part of the main point in this article.

**Suggestions for further research:** “The I/S success model needs further development and validation before it could serve as basis for the selection of appropriate I/S measures. In the meantime, careful attention must be given to the development of I/S success instruments”


**Motivator:** Only peac-meal, ad-hoc techniques are available for measuring and improving IQ in organization

**Problem:** “The research challenge is to develop an overall model with an accompanying assessment instrument for measuring IQ. Furthermore, techniques must be developed to compare the assessment results against benchmarks and across stakeholders. Such techniques are necessary for prioritizing IQ improvement efforts”…

**Empirical base:** One study testing an Information Quality Assessment (IQA) instrument. The article reports that 261 respondents in five organizations answered the questionnaire.

**Result:** The PSP/IQ model (organizes the key IQ dimensions), the IQA instrument (measure IQ) and the IQ gap analysis techniques (a tool for comparisons). According to the article .."each component of the AIMO has a merit in itself”…”AIMO is a management tool for conceptualizing and assessing IQ in business terms…a practical IQ tool to organizations”

**Practical application:** Implicit since the developed model is aimed for practical use.

**Suggestions for further research:** None


**Motivator:** A clarification article is needed since IT governance has become a business problem

**Problem:** Clarification is needed.

**Empirical base:** A literature study using 23 references.

**Result:** Referred tendencies under the IT audit “umbrella” (IT auditing, IT risk management, Cobit and ITIL), may help to measure the actual performance and quality of information systems and the business value of IT Governance initiatives.
**Practical application:** No.

**Suggestions for further research:** None.

## 5 Analysis

The most common result of the reading is a new or developed model. One model suggested six diagrams to be used together, another introduced a new software maintenance metric and also a matrix in the same article. A statement of the state of the art in the area was performed by two, and suggestions of using a specific model or improving the maintenance process was the result of three articles. One developed an ontology and one discussed IT Governance issues.

The articles where mainly based on literature studies, in one of them a small case was added to validate the result. In two cases the article was completely based on a case study. A survey was only performed in one article, and an in-depth interview (with three interviewees) was also presented in only one study. One study did a comparison of the software documentation, and observations of maintainers activities, and one did not mention what method they used at all. Only few of the articles included an in depth discussion on methods used.

Empirical data used in the articles were mainly collected from journals, conferences and standards. An example from a development project was used in a small case, and in the case of a survey data was collected from users and maintainers of software systems. In two cases existing models were improved, tried/validated, and in one case the ontology developed was used in three different projects/cases for validation. The in-depth interview was brought out with three researchers participating.

Six of the research articles did not suggest any topic for further research. The rest of the articles varied in the aim of suggestions for further research. One expressed the desire to test a metric on a live project, and another expressed a desire to develop a prototype based on the result. One wanted to plan and implement the ontology developed, and one wanted the model developed to be further validated and refined. Yet another stated the result and suggestion for further research to be an “invitation to research community to work together in this issue”. Complementing by in-depth interviews (and organisation analysis) was the ambition of one article, and finally one stated that the model and taxonomy developed will be used to guide future research for various reasons. The suggestions for further research are listed in table 3.

<table>
<thead>
<tr>
<th>Title</th>
<th>Suggestion on further research</th>
</tr>
</thead>
<tbody>
<tr>
<td>Types of software evolution and software maintenance</td>
<td>No suggestions</td>
</tr>
<tr>
<td>A Holistic Approach to Software Defect Analysis and Management</td>
<td>The authors state a desire to test the metric on a live project in all phases of software development process (from requirement to maintenance). A software prototype to automate defect fixation process will also be target for</td>
</tr>
</tbody>
</table>
The authors will focus on the stages of planning and implementation of the ontology in future research. This article only described the design of the ontology.

<table>
<thead>
<tr>
<th>Title</th>
<th>Suggestions</th>
</tr>
</thead>
<tbody>
<tr>
<td>An ontology for the management of software maintenance projects</td>
<td>The authors will focus on the stages of planning and implementation of the ontology in future research. This article only described the design of the ontology.</td>
</tr>
<tr>
<td>Information systems quality versus software quality - A discussion from a managerial, an organisational and an engineering viewpoint</td>
<td>No suggestions.</td>
</tr>
<tr>
<td>A Conceptual Model of Software Maintenance</td>
<td>Test and further development of the model.</td>
</tr>
<tr>
<td>Motivating the Corrective Maintenance Maturity Model</td>
<td>No suggestions.</td>
</tr>
<tr>
<td>Towards an Ontology of Software Maintenance</td>
<td>No suggestions.</td>
</tr>
<tr>
<td>Contributors to quality during software maintenance</td>
<td>No suggestions.</td>
</tr>
<tr>
<td>A comprehensive model for assessing the quality and productivity of the information systems function: toward a theory for information systems assessment</td>
<td>Additional research is suggested to advance the IS assessment contingency theory.</td>
</tr>
<tr>
<td>Information Systems Success: The Quest for the Dependent Variable</td>
<td>&quot;The I/S success model needs further development and validation before it could serve as basis for the selection of appropriate I/S measures. In the meantime, careful attention must be given to the development of I/S success instruments.&quot;…</td>
</tr>
<tr>
<td>AIMQ: a methodology for information quality assessment</td>
<td>No suggestions.</td>
</tr>
<tr>
<td>IT Governance Mechanisms in Managing IT Business Value</td>
<td>No suggestions.</td>
</tr>
</tbody>
</table>

Table 3: Suggestions for further research found in the article reading
The overall picture from the article reading is that empirical studies have had low priority in favor of development of new models and approaches that were common features of the articles studied.

6 Discussion and reflections

The purpose of this project was to analyze a limited number of articles in the areas of software maintenance, systems maintenance and IT-governance in order to find research questions and approaches for further research concerning quality IS. Our common experience of this analytic exercise can be divided into four areas of concern: Models, motivation and validation, definition uncertainty and use of concepts from research.

It is our impression that new models and approaches are suggested as solutions to a theoretically derived problem. Some examples of our reflections from our article reading:

...“If it is "premature to follow stages for individual organisations" in development and maintenance, then why are so many models developed?”...

...“There are many models developed in the area. Is there a need for more models and how general is the models presented? Test and validations of the models developed?”...

...“It seems more important to create a new and more comprehensive model than to contribute to practical utilization of previous developed models”...

Some questions were raised during the article reading e.g.: Model development seems important but who is entitled to develop the existing models? The model creator or any interested researcher in the IS research community?

Many of the articles motivate research efforts without reference to demand from practice and it is common that validation of concepts is supposed to be an activity in future research projects. Hevner et al (2004) argues that evidence of need from practice is an important precondition for research and that evaluation of concepts is required. We are confused to notice that this tradition seems have no root in the research we have in this study. Some of our reflections on this theme:

...”A good taxonomy and model but how influential is or has the model been?”...

...”No proof of demand from research or practice for this new model. In design science this is a precondition. No evaluation. Is there really a demand for benchmarking on IQ?”...

The definition problem made us change focus of our project but we still earned some experience concerning the need for some kind of definition standard. One comment from our documentation may illustrate this.
...“discuss quality a lot, but does not define it”...

We are unable to draw any conclusions on quality IS since the picture seems so complex and hard to catch. One conclusion though is that the area IS quality has a great deal of interest from research. Our search for relevant research question has so far landed in three preliminary questions/statements:

- There is a need for empirical motivated and empirical based research
- Models or approaches should build on previous work or add something new and or original
- Use previous invented definitions (preferably from some kind of general standard)

This article is based on a very limited sample of articles and we can of course not draw any valid conclusions from our study. A more complete literature study for instance in line with the principles outlined by Kitchenham (2004) would give a more valid result. It is however our believe that the three questions raised above are important to discuss in order to explore the questions on direction, robustness and utilization of research in information systems.

References


