PERGAMON

Information Systems 28 (2003)

www.elsevier.com/locate/infosys

# Improving the quality of data models: empirical validation of a quality management framework ☆

Daniel L. Moody[a,b,*], Graeme G. Shanks[c]

[a] Department of Computer and Information Science, Norwegian University of Science and Technology (NTNU), Trondheim N-7491, Norway
[b] School of Business Systems, Monash University, Melbourne 3800, Australia
[c] Department of Information Systems, University of Melbourne, Melbourne 3052, Australia

Received 17 August 2001; received in revised form 10 February 2002; accepted 15 March 2002

**Abstract**

This paper describes the results of a 5-year research programme into evaluating and improving the quality of data models. The theoretical base for this work was a data model quality management framework proposed by Moody and Shanks (In: P. Loucopolous (Ed.), Proceedings of the 13th International Conference on the Entity Relationship Approach, Manchester, England, December 14–17, 1994). A combination of field and laboratory research methods (action research, laboratory experiments and systems development) was used to empirically validate the framework. This paper describes how the framework was used to: (a) quality assure a data model in a large application development project (*product quality*); (b) reengineer application development processes to build quality into the data analysis process (*process quality*); (c) investigate differences between data models produced by experts and novices; (d) provide automated support for the evaluation process (the Data Model Quality Advisor). The results of the research have been used to refine and extend the framework, to the point that it is now a stable and mature approach.


*Keywords:* Requirements analysis; Data modelling; Entity relationship model; Quality assurance; Action research

## 1. Introduction

### 1.1. The importance of data model quality

The choice of an appropriate representation of data is one of the most crucial tasks in information systems development. Although data modelling represents only a small proportion of the total systems development effort, its impact on the quality of the final system is probably greater than any other phase [1]. The data model is a major determinant of system development costs [2], system flexibility [3], integration with other systems [4] and the ability of the system to meet user requirements [5].

The traditional thrust of software quality assurance has been to use "brute force" testing at the end of development [6]. However, Total Quality Management (TQM) approaches suggest

☆ Recommended by Professor P. Loucopoulos.

*Corresponding author. Department of Computer and Information Science, Norwegian University of Science and Technology (NTNU), Trondheim N-7491, Norway. Tel.: +47-7359-3354; fax: +47-7359-4466.

*E-mail addresses:* dmoody@idi.ntnu.no, dmoody@infotech.monash.edu.au (D.L. Moody), g.shanks@dis.unimelb.edu.au (G.G. Shanks).


PII: S0306-4379(02)00043-1

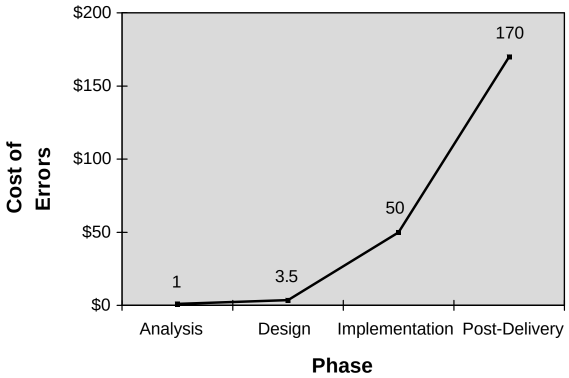


Fig. 1. Cost of errors by development phase.

that it is faster and cheaper to concentrate effort during the early development phases of a product, in order to detect and correct defects as early as possible [7]. According to Boehm [8], relative to removing a defect discovered during the requirements stage, removing the same defect costs on average 3.5 times more during design, 50 times more at the implementation stage, and 170 times more after delivery (Fig. 1). Empirical studies have shown that moving quality assurance effort up to the early phases of development can be 33 times more cost effective than testing done at the end of development [9].

This suggests that substantially more effort should be spent during early development phases to catch defects when they occur, or to prevent them from occurring altogether. However, it is during analysis that the notion of software development as a craft rather than an engineering discipline is strongest, and quality is therefore most difficult to assess. There are relatively few guidelines for evaluating the quality of data models, and little agreement even among experts as to what makes a "good" data model. As a result, the quality of data models produced in practice is almost entirely dependent on the competence of the data modeller [10,11].

### 1.2. Product vs. process quality

In the quality management literature, the distinction is frequently made between product and process quality [12]:

- *Product quality* focuses on the characteristics of the product. Product quality criteria are used to carry out inspections of the finished product and detect and correct defects. This is the traditional approach to quality assurance.
- *Process quality* focuses on the process used to produce the product. The objective is to build quality into the production process rather than trying to add it in at the end through reviews and inspections of the finished product. The focus of process quality is on defect *prevention* rather than *detection*, and aims to reduce reliance on mass inspections as a way of achieving quality [13]. This is the TQM approach to quality assurance.

In the context of data modelling, product quality is concerned with evaluating and improving the quality of the data model (the product) while process quality is concerned with improving the data analysis process (*the production process*) (see Fig. 2). Product quality is most important in the context of an individual project—it is important to ensure that the data model is free of defects so that a database can be built which will meet user requirements. However process quality is more important in the wider organisational context: to improve the organisation's ability to efficiently deliver high quality information systems.

### 1.3. Previous research on data model quality

Previous research on data model quality has focused almost exclusively on product quality. A summary of approaches to quality in data modelling is shown in Table 1.

The simplest type of quality evaluation approach is where quality is defined as a list of desirable properties of a data model (e.g. [1,14–16]). Such lists provide a useful starting point for understanding and evaluating quality in data models, but are mostly unstructured, use imprecise definitions, often overlap, and properties of models are often confused with language and method properties [17].

More comprehensive approaches to quality evaluation develop theoretical frameworks which define the key concepts underlying data model quality. Lindland et al. [17] define a framework based on semiotic theory, which defines a conceptual model as a set of statements in a language. For each semiotic level (syntactic, semantic, pragmatic) the framework defines quality goals and means to achieve them. Krogstie et al. [10] extend the framework to include a fourth semiotic level: the social level. These frameworks apply to conceptual models generally, not just data models. Kesh [18] develops a framework for evaluating data models based on ontological concepts. This framework defines criteria and metrics for evaluating the quality of data models.

### 1.3.1. Weaknesses of existing research

The most serious deficiencies in the existing literature are:

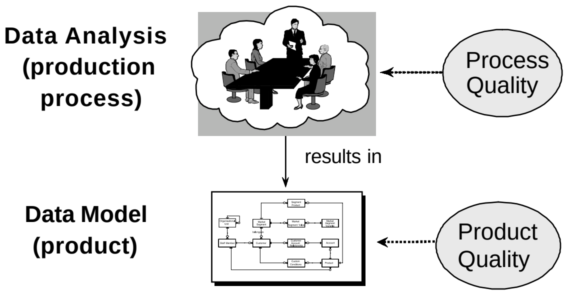


Fig. 2. Product vs. process quality.

- None of the approaches have been empirically validated in practice: all are either justified based on theory or the author(s)' experience. Theoretical justification is limited because methods have no "truth" value—the validity of a method is an empirical rather than a theoretical question [19,20]. Experiential justification is also limited because personal experience is subject to bias. Also, a method which works well for one person may not work for another [21].
- None of the approaches adequately addresses the issue of process quality: they define criteria and, in some cases, measures for evaluating the quality of data models (error detection) but not how to develop models in a high quality manner (error prevention).

Both of these issues are addressed in this paper.

## 1.4. Outline of the paper

The structure of the paper is:

- Section 2 describes the quality management framework used as the theoretical basis for this research—this represents the a priori theory being tested.
- Section 3 outlines the research methodology used to validate the framework.

Table 1
Approaches to data model quality

| APPROACH | COMPONENTS | TYPE |
|---|---|---|
| von Halle (1991) | Features of data models which maximise value to the organisation | List |
| Batini et al. (1992) | Quality features of a good schema, schema transformations | List |
| Levitin and Redman (1994) | Quality dimensions, reinforcements and trade-offs | List |
| Lindland et al. (1994) | Based in semiotic theory, separation of quality goals from means | Framework (semiotics) |
| Krogstie et al. (1995) | Extends Lindland et al's framework with agreement goal and social construction theory | Framework (semiotics) |
| Kesh (1995) | Separates ontology from behaviour. Defines metrics for evaluating quality. | Framework (ontology) |
| Witt and Simsion (2000) | Design and evaluation of alternative ER models | List |

- Section 4 describes how the framework was used to quality assure a data model for an application development project as part of an action research study (*product quality*).
- Section 5 describes how the framework was used to re-engineer the analysis process in an organisation as part of a longitudinal action research study (*process quality*).
- Section 6 describes how the framework was used to analyse differences in the quality of models produced by expert and novice data modellers using a laboratory experiment.
- Section 7 describes how the framework was used to provide automated support for the evaluation process (the Data Model Quality Advisor), and analyses its effectiveness using a laboratory experiment.
- Section 8 summarises the research findings and their implications for research and practice.

## 2. A framework for evaluating and improving the quality of data models

### 2.1. Overview of the framework

The quality management framework used as the basis for this research is defined by the Entity Relationship model in Fig. 3 [11]. This represents the a priori theory being tested by this research. The purpose of the framework is to evaluate and improve the quality of application data models. The framework consists of five major constructs, each of which is shown as a separate entity in Fig. 3:

- Quality factors define the characteristics of a data model that determine its overall quality. A particular quality factor may have positive or negative interactions with other quality factors (shown as a many-to-many relationship in the diagram). These represent the trade-offs implicit in the modelling process.
- Stakeholders are people who are involved in building or using the data model, and therefore have an interest in its quality. Different stakeholders will generally be interested in different quality factors.
- Quality metrics define ways of evaluating particular quality factors. There may be multiple measures for each quality factor.
- Weightings define the relative importance of different quality factors in a problem situation. These are used in making trade-offs between quality factors.
- Improvement strategies are techniques for improving the quality of data models with

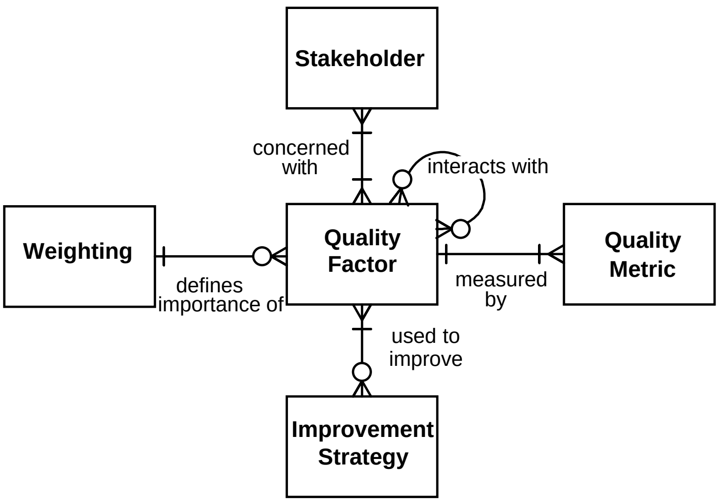


Fig. 3. Data model quality management framework.

respect to one or more quality factors. A particular improvement strategy may affect multiple quality factors.

This framework was not derived from theory, but emerged from many years experience in practice quality assuring data models. This represents "Mode 2" research as defined by Gibbons et al. [22], which is a new mode of knowledge production in which knowledge is generated in the context of application rather than from existing theory.

### 2.2. Stakeholders

The design of effective systems depends on the participation and satisfaction of all relevant stakeholders in the design process [109]. This includes both *upstream* participants (people who provide inputs to the data modelling process) and *downstream* participants (people who use the data model). The key stakeholders in the data modelling process are:

- Business user(s), whose information requirements are supposed to be represented in the data model.
- Data analyst(s), who are responsible for developing the data model.
- Data administrator(s), who are responsible for ensuring that the data model is consistent with the rest of the organisation's data.
- Application developer(s), who are responsible for implementing the data model (translating it into a physical database schema).

Each stakeholder role may be filled by multiple people, and the same individual may perform multiple roles. The data model acts as a communication vehicle among the various stakeholders.

### 2.3. Quality factors

The proposed quality factors and the stakeholders primarily interested in each are shown in the "fishbone" diagram in Fig. 4. Fishbone diagrams are widely used in quality management to show cause and effect relationships [12]. In this case, the diagram shows how each quality factor contributes to the overall quality of the data model. Together the set of quality factors incorporate the needs of all stakeholders, and define a complete picture of data model quality. Quality factors may be used as criteria for evaluating the quality of individual data models and comparing alternative representations.

The definitions of the quality factors are:

- Completeness refers to whether the data model contains all user requirements.

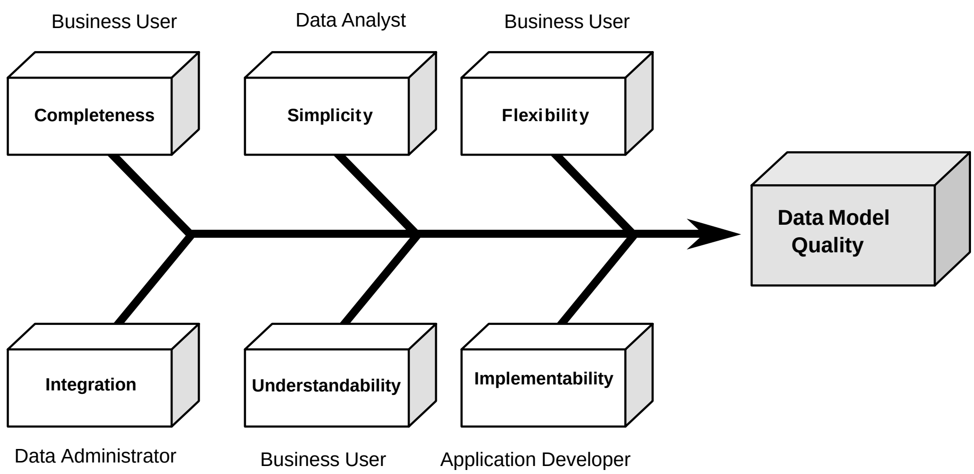


Fig. 4. Data model quality factors.

- Simplicity means that the data model contains the minimum possible entities and relationships.
- Flexibility is defined as the ease with which the data model can cope with business and/or regulatory change.
- Integration is defined as the consistency of the data model with the rest of the organisation's data.
- Understandability is defined as the ease with which the concepts and structures in the data model can be understood.
- Implementability is defined as the ease with which the data model can be implemented within the time, budget and technology constraints of the project.

For more detailed definitions of these quality factors, see Moody and Shanks [11].

### 2.4. Metrics

A range of metrics (29 in total) have been defined for evaluating the quality factors. These are described in detail in [23].

### 2.5. Improvement strategies

Improvement strategies were not defined in detail in the original formulation of the framework (they were defined in intension rather than extension), but were populated as part of the empirical validation process (see Section 5).

## 3. Research methodology

### 3.1. Validation of IS design methods

The question of how to validate IS design methods has been a longstanding issue in the IS field (e.g. [19,21,24–28]). There are inherent problems evaluating any methodology or design technique since there is typically no theory, no hypotheses, no experimental design and no data analysis to which traditional evaluation criteria can be applied [28].

As a result, IS design research tends to emphasise the development of new design methods and frameworks, while addressing the use and evaluation of methods in practice in only a limited fashion [21,27,29–32]. Wynekoop and Russo [21] conducted a review of IS design research published in the leading IS journals over the past three decades. The results of the analysis showed a heavy reliance on normative research, largely focusing on the development of new methods or modifications to existing methods. Wynekoop and Russo concluded that there was a "lack of serious empirical research into the efficacy of methods in practice" and a "need for validation of methods in organisational contexts using real practitioners".

A possible reason for the lack of validation of IS design methods is the philosophical and methodological difficulties involved in validating methods as opposed to theses. According to Rescher [33], human knowledge consists of two types:

- *Theses* or "knowledge that": these define statements or assertions about the world.
- *Methods* or "knowledge how": these define ways of doing things.

"Knowledge that" has been the major focus of scientific research, which is generally about establishing the truth of particular propositions (*hypotheses*). Rescher argues that an entirely different approach is required to validate methodological knowledge than to validate theses. The reason for this is that methods have no truth value, only *pragmatic value*—a method cannot be true or false, only effective or ineffective. Factual theses can either be established deductively from other theses or inductively from observations. The validity of a method can only be established by applicative success in practice.

### 3.2. Research design

There are a wide variety of research methods which may be used in conducting IS research [21,34–38]. Different research methods are appropriate in different situations, depending on the research question and the stage of knowledge in the area being studied [21,35,38]. In general, a combination of research methods may be most effective in achieving a particular research objective [21,27,36,39–42]. For example, when a subject area is not well understood, qualitative methods may be used to build theory and testable hypotheses. Theory may then be tested using quantitative methods such as surveys and experiments.

In this paper, a combination of field and laboratory, quantitative and qualitative research methods are used to validate the framework:

- Action research (field based, qualitative),
- laboratory experiment (laboratory based, quantitative),
- systems development (laboratory based, qualitative).

Fig. 5 summarises the research design.

A field-based qualitative method (action research) was used in the first two phases, to evaluate the framework in a real world context. Using this approach, the framework has been applied in two of the largest commercial organisations in Australia. A laboratory-based quantitative method (laboratory experiment) was then used to evaluate differences in data models developed by experts and novices. The final research phase used a mixed method approach: systems development was used to develop an automated tool (the Data Model Quality Advisor) to support the framework and then a laboratory experiment was used to evaluate the effectiveness of this tool. Mixing qualitative and quantitative research methods is called *triangulation of method* [42]. The two types of methods have different, complementary strengths and when used together can lead to a more comprehensive understanding of a phenomenon [39,40].

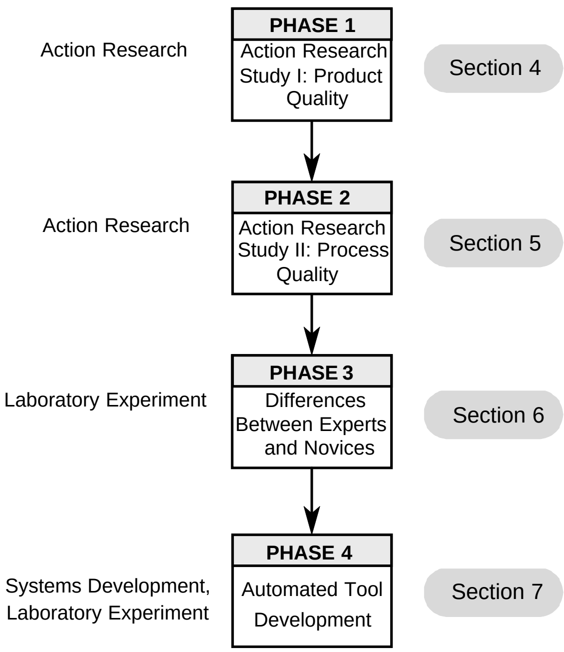


Fig. 5. Research design.

## 4. Action research study I: product quality

This section describes how the framework was used to evaluate and improve the quality of a data model in a large application development project. This was the first real world application of the framework. In this case, the framework is used to improve *product quality*.

### 4.1. Action research

A major barrier to the empirical validation of IS design methods is that it is very difficult to get new approaches, especially those developed in academic environments, accepted and used in practice. Practitioners who have developed familiarity and expertise with existing techniques are reluctant to adopt academic approaches that are theoretically sound but unproven in practice [21,29,43]. This paper uses *action research* as a way of overcoming these barriers.

Action research is an alternative social science research approach which links theory and practice to solve practical problems in the field [34,44–46]. It has a long history of successful application in other applied disciplines, such as education, psychology and health care [47], and can be applied in field settings where more traditional experimental or quasi-experimental methods cannot easily be applied [48]. One of its major advantages is that it can help to overcome the problem of persuading practitioners to adopt new techniques, and overcome the cultural divide that exists between information systems academics and practitioners [31,43,49,50].

Despite the clear applicability of action research in an applied discipline like IS, there has been remarkably little research of this kind in the IS literature [43]. For example, a survey of IS design research over the past three decades by Wynekoop and Russo [21] found that action research accounted for less than 4% of research papers published. Lau [51] found only one action research article in a 25-year literature review covering four mainstream IS journals.

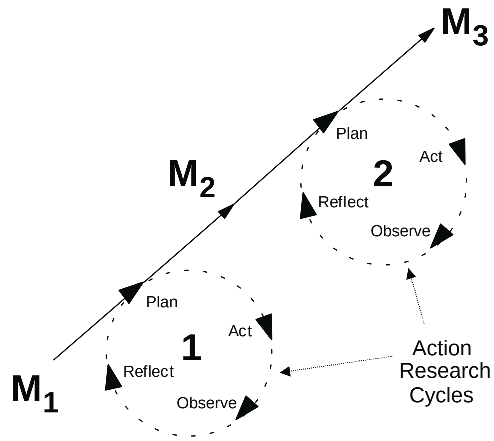


Fig. 6. The action research spiral (adapted from Hatten et al. [56]).

*4.1.1. The action research process*

The originator of action research is usually taken to be Kurt Lewin, an American psychologist [49,52]. In the 1940s, Lewin constructed a theory of action research, which described action research as proceeding in a "spiral" of steps, each of which is composed of planning, action and the evaluation of the result of the action [53]. This formalisation of action research theory made action research an acceptable method of inquiry [45]. Each action research cycle consists of the following steps [54] (see Fig. 6):

- Plan: develop a plan of action to improve current practice. The plan must be flexible to allow adaptation for unforeseen effects or constraints.
- Act: the participants act together to implement the plan.
- Observe: the action is observed to collect evidence which allows thorough evaluation of outcomes. A variety of data collection methods may be used to evaluate the results of the intervention [46,55].
- Reflect: group members reflect on what went wrong, what went right and how to improve the idea in the next cycle. All participants may contribute to the refinement of the idea. This provides a basis for further planning of critically informed action, thereby continuing the cycle.

Each cycle may lead to improvement of the original idea ($M_1$), resulting in a sequence of successively refined and improved ideas $M_2, M_3 \ldots$.

*4.1.2. Strengths and weaknesses of action research*

The strengths of action research are:

- It is a field-based method, which allows testing of the framework "in an organisational context using real practitioners", as recommended by Wynekoop and Russo [21]. In IS design research, very few methods are ever applied outside of a research environment [21,29,27,50].
- Because it is action oriented, it is particularly suitable for validating "knowledge how" (methods).
- It involves collaboration between researchers and practitioners. There is quite a wide gap between IS research and practice [50,57–60], and action research provides a mechanism for knowledge transfers between the two groups.
- It allows research ideas to be refined via of an iterative learning process, so is highly appropriate in exploratory research [61]. Traditional hypothesis testing methods such as experiments are less applicable in exploratory research as they typically result in yes/no answers to questions, and provide little feedback as to how to improve the idea [62].

The weaknesses of action research are:

- Difficulty of generalising results beyond specific cases: like case study research, action research is concerned with single situations, from which it is difficult to generalise.
- Researcher bias: the researcher sacrifices some level of objectivity in action research by being directly involved in the action.
- Lack of control: it is never possible to be sure that the outcomes achieved were due to the intervention and not some other factor.

A range of strategies may be used to address these potential weaknesses [63].

### *4.2. Background to the situation*

The organisation involved in this study was a telecommunications company, and one of Australia's largest commercial organisations. The organisation had embarked on a major application development project to replace the core operational systems in its telemarketing centre. This project was regarded as mission critical, and as a result, an independent review was commissioned at the end of the requirements analysis stage to confirm that the data model would meet current and future business requirements. The consultancy was offered as part of a competitive quotation process.

### *4.3. The intervention*

The quality management framework described in Section 2 was used to conduct a quality review of the data model. The review was conducted over 10 working days, and involved review sessions with the project team, inspection of the model, interviews with business representatives, application development staff and Corporate Data Management staff. A report was produced with separate sections for each quality factor. Over 100 individual quality issues were identified (error detection) and a number of recommendations were made for improving the model prior to proceeding to the design phase (error correction).

Fig. 7 summarises the results of the review in the form of a Kiviat chart. Each quality factor was rated on a scale of 1 to 5 (5 = Excellent; 1 = Poor). The chart shows that the model was technically sound (correctness) but had major deficiencies in completeness, flexibility and integration. (Note: correctness was a new quality factor introduced as part of this study.)

### *4.4. Action outcomes: benefits of the framework*

One of the unique characteristics of action research compared to other research methods is that it results in practical benefits for the organisation being studied (*action outcomes*) as well as the discovery of new theoretical knowledge

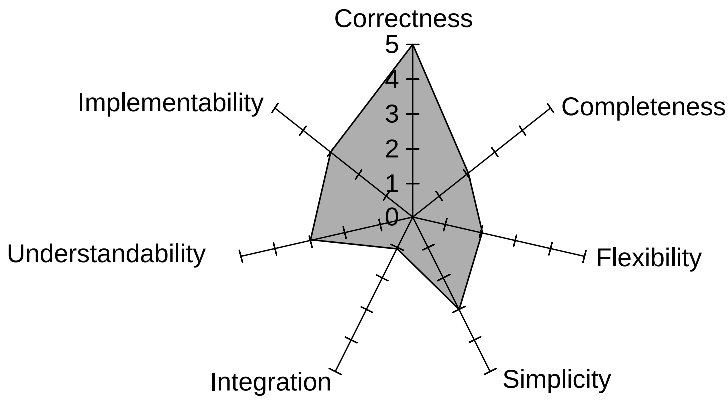


Fig. 7. Results of evaluation.

(*research outcomes*). Here we summarise the action outcomes of the study.

*4.4.1. Systematic approach*

The framework provided a systematic approach to conducting the review. Use of clearly defined evaluation criteria (quality factors) helped to focus the task and carry out a comprehensive analysis in a very limited timeframe. In the absence of the framework, the review would have been much more ad hoc, and many issues may have been overlooked.

*4.4.2. Credibility of findings*

Use of an explicitly defined and internationally published quality framework gave considerable credibility to the findings of the review. In particular, it gave management confidence that the review had been conducted in an objective manner and that all aspects had been covered. The project manager also said that inclusion of the framework in the consultancy proposal was one of the major reasons for it being selected as the successful bid. None of the other consulting firms involved in the quotation process had an explicit framework for conducting the review and relied instead on expert opinion.

*4.4.3. Development cost savings*

A number of the quality issues identified in the review led to direct or indirect cost savings for the project:

- Missing requirements: a number of instances were found where user requirements were omitted from the model—these represent errors of *completeness*. Correcting these errors led to indirect cost savings, as the cost of adding these requirements later on in the lifecycle would have been many times higher (3.5–170 times the cost).
- Unnecessary requirements: a number of instances were found where the data model included requirements that users had not asked for or were out of scope—these also represent errors of *completeness*. In most cases, these represented supposition on the part of the project team about what users *might* want rather than what they actually did want. Removing these requirements directly reduced development costs.
- Overlap with existing systems: a large number of instances were found where data stored in existing systems was included in the model—these represent errors of *integration*. Removing this duplication reduced the size of the model by almost half, which directly reduced development costs, as well as preventing ongoing costs of data duplication (storage, data entry, reconciliation and synchronisation costs) in the future [4]. This was the most significant area of cost savings identified by the review.
- Inconsistencies in data definitions: a number of instances were found where data items were defined inconsistently with how they were defined in other corporate systems with which the new system needed to interface. Resolving these inconsistencies led to indirect savings in interface development costs and ongoing data translation costs.

*4.4.4. Reduction in data maintenance costs*

An unexpected benefit of the review, and one which extended beyond the project into general work practices, was the reduction in manual data maintenance effort. In identifying overlap between the data model and other corporate systems (integration), it was discovered that the business unit was maintaining a number of data sets that could have been sourced from other areas. For example, it was a full-time job for one person to maintain up-to-date information about telephone exchange capabilities. However this data was also being maintained by engineering staff responsible for installing and repairing exchange equipment. This was the most accurate source of this information, as it was updated as soon as equipment was installed or upgraded. By obtaining this data directly from its source rather than maintaining copies, this data maintenance overhead was eliminated. This, together with other similar cases, resulted in estimated cost savings for the organisation of over $100,000 per annum.

*4.4.5. Involvement of stakeholders*

A major benefit of using the framework was that it explicitly involved all stakeholders in the review process. This highlighted problems in the lack of stakeholder involvement and differences in stakeholder perspectives. Most other quality frameworks focus only on the characteristics of the model itself. The stakeholder concept helps to focus also on people issues, which are important in understanding the root causes of many quality problems. In this case, the areas where the model was most deficient (completeness, flexibility and integration) were due primarily to the lack of involvement of the relevant stakeholders. The project team had been carrying out their analysis largely in isolation—users had been involved in only a very limited capacity, and application developers and data administrators had not been involved at all. There was also a difference in perspectives between the analysts and business users: the analysts saw the project primarily as a technical reengineering exercise, while business users saw the new system as providing them with new capabilities to compete.

### *4.5. Research outcomes: learning about the framework*

Here we summarise the changes to the framework as a result of this study.

#### *4.5.1. Validation of framework constructs*

The empirical validity of four of the five framework constructs was evaluated as part of this study. The results were mixed:

- Quality factors: these were found to be very useful as they defined "what" to evaluate. Having a clearly defined set of evaluation criteria helped to focus the review process and forced the discipline of considering all aspects of quality.
- Stakeholders: these were also found to be useful, as they defined "who" to involve in the review process. The lack of stakeholder involvement emerged as the root cause of most of the quality problems in the model.
- Metrics were sparingly used. Given the limited timeframe allowed for the review, there was only time to apply a small subset of the metrics, so this was not a very thorough test of their usefulness.
- Weightings proved problematic to apply in practice. The project team found it difficult to agree on weightings for quality factors because some factors were important in some parts of the model but not in others. For example, flexibility was considered highly important in the area of marketing campaigns which was critical to the organisations ability to compete, but less so in the area of rostering, where requirements were stable and unlikely to change.

Improvement strategies had not yet been defined in detail, so could not be validated as part of this study. However this study formed the basis for initial population of this construct.

#### *4.5.2. Validation of quality factors*

The set of quality factors were empirically validated by mapping quality issues raised in the review against the quality factors defined. The three issues that need to be considered in validating the proposed set of quality factors are [64]:

1. *Sufficiency:* is the set of quality factors sufficient for evaluating the quality of data models—that is, is the set of quality factors complete? This will be validated if all quality issues raised relate to at least one of the quality factors defined.
2. *Necessity:* are all quality factors necessary for evaluating the quality of data models—that is, are all quality factors relevant? This will be validated if each quality factor has at least one relevant issue identified in the review.
3. *Independence*: are the definitions of the quality factors independent of each other? This will be validated if each quality issue relates to at most one quality factor.

The validation process is illustrated in abstract in Fig. 8. Actual issues raised (empirical observations) are mapped against the proposed quality factors (a priori theory). In the diagram:

- Issue no. 1 does not map to any quality factor: this shows that the set of quality factors is incomplete.

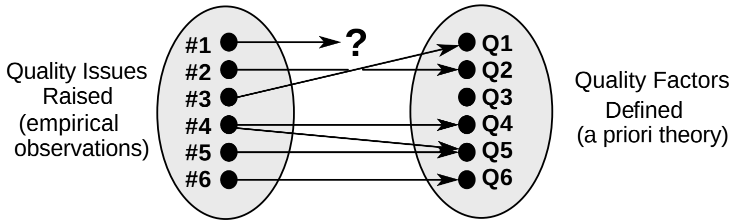


Fig. 8. Empirical validation of quality factors.

- No quality issues map to quality factor Q3: this shows that this quality factor may be unnecessary.
- Issue no. 4 maps to two quality factors: Q4 and Q5: this shows that the definitions of these quality factors overlap.

Of course, it could be argued that because the set of quality factors was used as a basis for conducting the review, this is not an independent test of the validity of the quality factors. However review participants were asked to identify all relevant quality issues and not to be limited by the quality factors defined.

*Sufficiency*: one new quality factor, *correctness*, was added to the set of quality factors as a result of quality issues raised that did not map to any of the quality factors. Correctness was defined as whether the model conforms to the rules of the data modelling technique (i.e. whether it is a *valid* data model). This includes diagramming conventions, naming rules, definition rules, rules of composition and normalisation.

*Necessity*: all quality factors had at least one quality issue identified, which indicated they were all relevant. There was also agreement among all stakeholders that all quality factors were determinants of the quality of the model. Interestingly, the quality factor which led to most of the benefits in this study—*integration*—does not appear in most frameworks previously proposed in the literature. Most approaches to data model quality consider a data model as a standalone artifact (closed system), whereas in practice, an information system forms a small part of a much larger information processing environment (open system). While a data model may have a high level of quality in its immediate context, it may have a poor fit with other systems. Lack of integration leads to problems of data duplication, complex interfaces and problems consolidating data from different systems, which can have high ongoing costs for the organisation [4]. The need to consider individual systems in the context of an overall architecture is critical for developing quality information systems [65].

*Independence*: a number of issues raised in the review could have been classified under multiple quality factors, indicating overlap between their definitions. For example, the inclusion of unnecessary requirements could have been classified as errors of completeness, as they represented a mismatch between the model and user requirements. Alternatively, they could have been classified as errors of simplicity, as they added unnecessary complexity to the model. The definitions of all quality factors were refined to make the distinctions between them clear.

Fig. 9 shows the revised set of quality factors as a result of this study.

#### *4.5.3. Process quality issues*

While the review focused on product quality, most of the flaws in the model could be traced back to problems in the process used to develop it. The analysts involved were very technically competent (the model scored a perfect “5” on correctness), but their major failing was in not involving the other stakeholders in the process. This was the root cause of the problems where the model was most deficient: completeness and flexibility (business users), integration (data administrators) and implementability (application developers). For this reason, a major recommendation

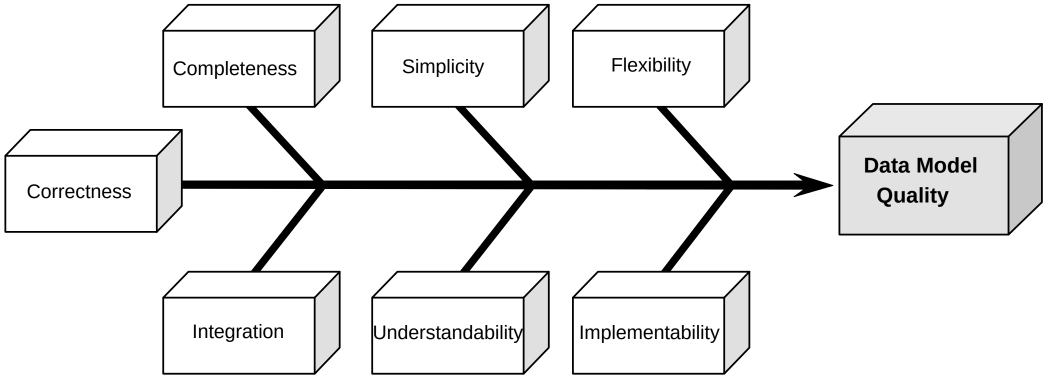


Fig. 9. Revised set of quality factors.

was to change the process to explicitly involve all stakeholders. This highlighted a major weakness of the framework—while it addresses product quality (how to evaluate and improve the quality of the finished product), it says little about process quality (how to develop high quality data models). As such, it is not a true quality management framework but a quality assurance framework: it reflects a focus on error detection and correction (the traditional approach to quality assurance) rather than error prevention (the TQM approach).

### 4.6. *Strengths and weaknesses of the research*

#### 4.6.1. *External validity*

This study was conducted in a mission critical project in a large and complex organisation, so represents a genuine "real world" test of the method. It therefore has high face validity. However, the generalisability of findings from a single project in a single organisation is clearly limited. In general, multiple cycles and case studies are required to increase generalisability of results [63].

#### 4.6.2. *Internal validity*

Action research has features that make it possible to achieve high levels of rigour—higher sometimes than quasi-experimental methods can achieve in the same setting [66]. The following strategies were used to improve the internal validity of the results.

*Active seeking of disconfirming evidence*: this is the primary method used to overcome potential researcher bias (e.g. selective observation) and involves actively seeking evidence that contradicts the expected outcomes. The objective of any action research project should not be to show that the idea works but to look for weaknesses and to improve it. In each cycle, the researchers challenge the emerging conclusions by looking for exceptions and trying to disprove the a priori theory [63]. In this study, both positive and negative feedback was sought from participants on the framework.

*Use of multiple informants*: in this study, the views of all relevant stakeholders (analysts, end users, database designers and data administrators) were sought regarding the effectiveness of the framework. This corresponds to the concept of *triangulation of sources* [42].

*Implementation of recommendations*: while the interpretation of the results of an intervention is by nature qualitative and subjective, the fact that the organisation implemented the recommendations of the review provides objective evidence that the framework produced useful results. The client's willingness to act on the results of an action research study is an indication of the validity of the results [67].

## 5. Action research study II: process quality

This section describes how the framework was used to improve the process of developing data models as part of a longitudinal action research study in a single organisation [31]. One of the principles of TQM is that the most effective way to improve the quality of a product is to improve the process by which it is developed [13]. This was also one of the major findings from the first action research study.

### *5.1. Background to the situation*

The organisation involved in this study was a large Australian bank, with over 25,000 staff. A central Data Administration (DA) group was responsible for developing and/or reviewing data models produced by application development projects. The organisation had a standard systems development methodology and a common corporate repository for storing all data models. The methodology prescribed that the DA group reviewed all data models at the end of the requirements definition stage, after they had been signed off by business users. Sign-off by the DA group was required before projects could progress to the logical design stage. Sign-off was also required at the end of the logical design phase to ensure conformance between the database design and data model.

A number of problems were identified with the existing process as a result of a review of practices and of data models stored in the corporate repository:

- Data duplication: the most serious problem identified as a result of the review was a high degree of overlap between different application data models. Different project teams were defining the same data in different ways, resulting in data redundancy and duplicated development effort. As an extreme case, 60 separate occurrences were found of the same attribute (Branch Number) in different data models. While the corporate repository was designed to facilitate data sharing and reuse, this was clearly not happening.
- Inconsistent quality of models: inspection of a sample of data models showed wide variations in quality. The major cause of this was large differences in levels of skill and experience within the group. While the policy was to have experienced data modellers working with inexperienced ones, lack of resources often prevented this. There was also no formal process for assuring the quality of data models produced by analysts within the DA group. While they had an important QA role in the systems development process, they had no internal QA process.
- Need for rework: in a large number of cases, project teams submitted data models for DA signoff at the end of the requirements definition stage, only for major quality problems to be found. This led to rework, delays to projects, added development costs and consequent friction between the DA group and project teams. In many cases, incorrect or poor quality models were signed off to enable projects to meet their deadlines. Most application development staff felt that the DA group was a hindrance to projects, and a major bottleneck in the development process.
- Database design problems: in a number of cases, large variations were found between the data model and the database design at the end of the logical design phase. This led to conflict between the DA group and the Database Administration (DBA) group. The DBA group argued that many data models signed off by the DA group could not be realistically implemented, and had to be heavily denormalised.

### *5.2. The intervention*

The data model quality framework was used as the basis for re-engineering the process of developing data models. The objective was to build quality into the data analysis process (*process quality*), rather than using the framework to evaluate the quality of data models at the end of the process (*product quality*) as in the previous study. This requires a focus on error prevention rather than error detection. The major changes introduced to the existing process were:

#### *5.2.1. Information architect role*

The most serious problem identified in the existing process was the degree of overlap between data models. The cause of this was that data analysts modelled the requirements of particular applications in relative isolation of each other. There was no one with responsibility for the "big picture", and hence little or no central coordination of their work. Despite the existence of a common corporate repository, data analysts were generally only aware of opportunities for reuse within their own experience (other projects they had worked on). This reflects that data sharing is primarily a behavioural rather than a technical issue [68]. High levels of data reuse are not automatic when a common repository is used—it requires careful management. To address this issue, a new position (Information Architect) was created within the DA group, with explicit responsibility for integration of data models across the organisation. The Information Architect was not involved in developing data models, only in reviewing them for consistency and overlap.

### 5.2.2. Preventative reviews

*Requirements definition phase*: traditional quality systems take a "big bang" approach to quality assurance, with an inspection carried out at the end of each phase to ensure conformance to specifications. This was the situation in the current quality assurance process, and resulted in a "no win" situation for both parties: projects were often delayed and the DA group was blamed for it. The concept of *preventative reviews* in TQM prescribes that iterative reviews should be carried out at specific checkpoints within each phase of development [7,69]. Regular reviews minimise rework at the end of phase, and allow reviewers to have active input into the model while it is being developed, rather than only at the end. Three review checkpoints were defined in the requirements definition phase (Fig. 10).

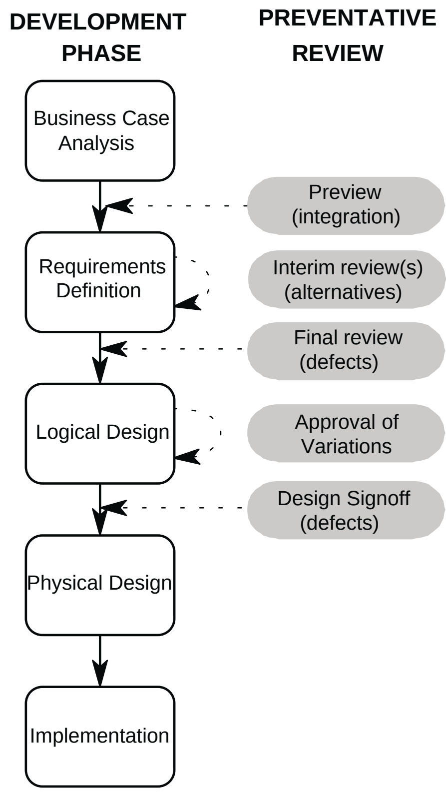


Fig. 10. Revised quality assurance process.

1. Preview: this review took place before any detailed analysis had been carried out, but after the scope of the project had been defined. The major objective of this review was to identify opportunities for reuse and sharing of data (*integration*).
2. Interim review(s): a review was then carried out at the midpoint of the requirements definition phase, when all requirements gathering activities (interviews, workshops, etc.) had been completed and a first cut model had been produced. The objective of this review was to explore alternative solutions, using the first cut model as a starting point. Sometimes multiple reviews were required in order to consider a reasonable range of alternative solutions. The quality factors considered in these reviews were completeness, flexibility, understandability, simplicity and implementability.
3. Final review (signoff): this was the stage of final approval, and if the earlier reviews were conducted properly, was largely a formality. The major focus of this review was correctness.

The participants involved in each review were the data analyst who developed the model, business user(s), the information architect, application developer(s) and a data analyst from outside the project. Each review participant was required to independently inspect the data model prior to the review and rate it on a scale of 1 to 5 on each quality factor. Quality metrics were calculated by the data analyst who developed the model. The review session then served as a forum for reaching convergence on evaluations and looking for ways to improve the model. In this way, all stakeholders were actively involved in developing the data model (as opposed to a passive review role).

*Logical design phase*: during the logical design phase, variations to the data model were approved by the project data analyst as they occurred (Fig. 10). Rather than calling a formal meeting for each variation, they were handled on a case-by-case basis, similar to how change requests are handled. Variations were agreed between the data analyst and database design, with the information architect used as the final arbiter if they could not agree. The purpose of this was to ensure that all major issues in the transformation of the data model to a database design were resolved prior to the final design review. The final design review involved the project data analyst(s), database designer(s) and the information architect, and was largely a formality.

### 5.3. Action outcomes: benefits of the framework

The process described above was used to quality assure over 20 data models over a 2-year period. A number of practical benefits were identified as a result of the intervention.

#### 5.3.1. Increased reuse and reduced rework

By far the most useful change to the data analysis process was the introduction of the "preview" review. While it may seem unusual (even contradictory) to have a quality review before any work has been done, this was where most of the benefits were achieved. By meeting with project teams before they had begun analysis, the DA group was able to save them work (rather than creating rework as in the existing process) by identifying opportunities for reuse of data. Over the 2-year period, average reuse levels more than doubled, from less than 20% (historical analysis showed an average reuse of data of 18.8%) to over 40% per project (41.3% in the second year). Actual savings in development costs as a result of data reuse are difficult to estimate, but empirical studies have shown that, in general, 1% reuse leads to a 1.9% saving in development costs [5]. This suggests that development cost savings were substantial—in fact, multiplying the increase in reuse by 1.9 results in a saving of 42.8%.[1]

#### 5.3.2. Reduced requirements errors

Errors or omissions in the data model lead to expensive changes in later development stages, resulting in rework and added cost for projects. This was a major source of project overruns and missed deadlines in the organisation. Data model related change requests submitted after the end of the Requirements Definition stage were reduced by almost 30% over the 2-year period. This resulted in estimated savings of over $1.2 million over the period (based on the average cost of dealing with data model related change requests).

#### 5.3.3. Innovative solutions

In the existing process, it was generally too late to explore alternatives by the time the model was reviewed. The consequences of changing the model at the end of the requirements definition phase were rework and delays to the project. As a result, project teams often became highly defensive, creating an environment which was not conducive to new ideas. Introduction of preventative reviews helped to change the focus of reviews from detecting errors (*negative quality*) to improving models and exploring alternative solutions (*positive quality*). The involvement of review participants external to the project team also helped to introduce different points of views and encouraged development of innovative solutions [70].

[1] Note that only reuse of data is considered here, so these are estimates only of database design cost savings. Savings in programming costs would also be expected as a result of data reuse, but are difficult to estimate—in many cases, the same data may be used, but new programs required to access and process the data.

#### *5.3.4. Increased awareness of stakeholder perspectives*

In the existing process, separate reviews were held for user signoff and DA signoff. This sometimes led to problems where a model was signed off by users and subsequently changed as a result of DA review. In the new process, all stakeholders were involved in each review and agreed on a single model which would proceed to the next phase. While this took longer than holding separate reviews, it was perceived to be worthwhile in the long term in improving different stakeholders' awareness of other stakeholders' perspectives and thereby improving working relationships.

#### *5.3.5. Continuous improvement/organisational learning*

Collection of metrics on the occurrence of defects helped to identify patterns of defects which could be addressed by preventative measures such as training or process change. In the existing process, each project was handled on a case-by-case basis, with organisational learning from each experience. While individual analysts learnt from each project, this did not translate into process improvement.

#### *5.3.6. Improved transition to database design*

In the existing process, application developers did not see the data model until the end of the requirements definition stage, after it had been signed off by the DA group. This frequently resulted in models that designers felt were impractical, and which needed to be changed drastically. TQM recommends that both *upstream* and *downstream* participants should be involved in each phase of development, in order to highlight potential problems as early as possible in the development lifecycle [13,69]. Involvement of application developers in reviews during the requirements definition phase was found to be beneficial for several reasons:

- It allowed them to gain familiarity with the model prior to the design stage to ensure a smooth transition,
- it gave them the opportunity to flag any potential implementation issues,
- it provided a "reality check" on what was technically possible and/or economically feasible,
- it reduced the number and severity of variations introduced during the logical design phase.

In this way, use of the framework led to improvements in a downstream process (database design).

#### *5.3.7. Improved quality and consistency of models*

In the revised process, at least two experienced data modellers (the Information Architect and one other) were involved in developing each model through the preventative review process. This resulted in better leverage of experienced resources and significantly improved the quality and consistency of models produced. The review process allowed inexperienced data modellers to learn from more experienced members of the group and thus facilitate skills transfer.

### *5.4. Research outcomes: learning about the framework*

Here we summarise the changes to the framework as a result of this study.

#### *5.4.1. Validation of framework constructs*

The empirical validity of all framework constructs was evaluated in this study. Two new constructs were added and one was removed as a result.

*New construct: quality review*: the most important research outcome of this study was the extension of the framework to include the process dimension. Previously, the framework was focused exclusively on product quality—how to evaluate the quality of a finished data model (error detection). In this study, the framework was augmented to incorporate process quality aspects—how to build quality into a data model as part of the data analysis process (error prevention). This required the introduction of a new construct called Quality Review, which defines the concept of preventative reviews. Each review checkpoint defines:

| NO. | ISSUE DESCRIPTION | QUALITY FACTOR | PRIORITY | STATUS | RESOLUTION |
|---|---|---|---|---|---|
| 1. | Definition of Customer does not include external clients, only internal business units | Flexibility | 1 | Resolved | Definition of Customer expanded and subtypes introduced for internal and external clients |
| 2. | Little information in common between subtypes of Communication Result | Correctness | 3 | Resolved | Subtypes remodelled as separate entities and supertype removed |
| 3. | Staff training and skills information (e.g. language capabilities) missing | Completeness | 1 | Open | Will require additional analysis work |

Fig. 11. Issues matrix.

- who: stakeholders involved in the review,
- what: quality factors considered in the review,
- when: where in the development process each review takes place.

*New construct: quality issue*: an unexpected finding of this study was that "soft" information in the form of textual descriptions of quality issues, was perceived by project teams to be the most valuable output of the review process. This information provided the major basis for improving the model—each issue represents a defect which needs to be corrected. Fig. 11 shows an example matrix used for recording quality issues. Each issue is classified by quality factor, prioritised on a scale of 1 (critical), 2 (important) or 3 (desirable) and resolutions recorded. This information could be used as the basis for developing *design rationale* explanations of the model [71].

Given the practical importance of this information, the framework was extended to include Quality Issue as an explicit construct. Each issue is classified by a particular quality factor. The classification is used to identify improvement strategies and to track patterns of defects over time.

*Removed construct: weighting*: the Weighting construct was removed from the framework as it was found to be problematic to apply in practice. As identified in the first study, the importance of a particular quality factor is not homogeneous throughout a data model. Therefore assigning an overall weighting to a quality factor is not meaningful. The concept of weightings was more applicable at the level of individual quality issues as described above, in order to prioritise improvements to the model. However this did not seem to justify an explicit construct.

The revised evaluation framework is shown in Fig. 12 (new constructs shaded).

### *5.4.2. Validation of quality factors*

Two new quality factors were proposed for inclusion in the framework as part of this study:

- Redundancy: a major focus of data modelling in practice is to ensure that the model contains no redundancy—that each fact is represented in a single place [1]. This was included as part of correctness as inclusion of redundancy can be considered to be an error in the application of the data modelling technique. However given its importance in data modelling, it could have been defined as a quality factor in its own right. The resolution was to leave it as part of correctness, because most analysts felt that it was part of the technical correctness of the model.
- Integrity: definition of business rules (integrity constraints) was originally included as part of completeness, as they form part of user requirements. However business rules are particularly important in a financial environment, because of the need to guarantee data integrity and enforce policies. Also, there has been increasing emphasis in practice on the use of business rules as a requirements gathering technique [72,73]. Consequently, it was decided

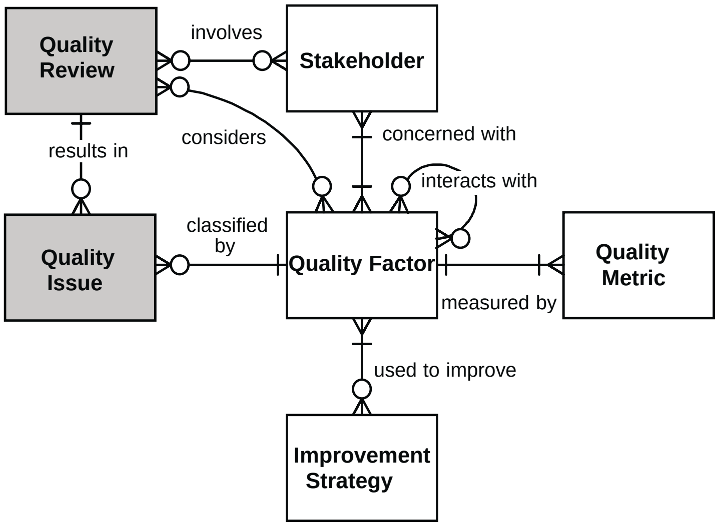


Fig. 12. Revised data model quality evaluation framework.

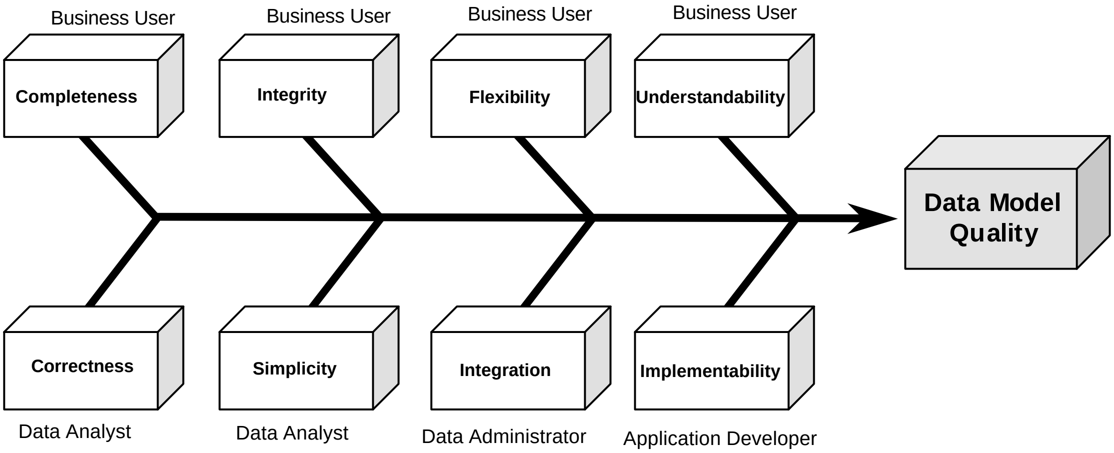


Fig. 13. Revised quality factors.

to extend the framework to include integrity as a quality factor in its own right. In terms of the 100% principle [74], this separates static aspects (completeness) and dynamic aspects (integrity).

The final set of quality factors, together with the stakeholder primarily interested in each, is shown in Fig. 13. The number of quality factors increased from six to eight as a result of the two action research studies (correctness was added as a result of the first study and integrity as part of the second study), which is still manageable. Psychological studies using a wide range of different stimuli have found that the limits of human cognitive capacity is "seven plus or minus two" concepts at a time [75,76]. For this reason, it is desirable that the number of quality factors be kept to less than 10.

### *5.4.3. Validation of metrics*

Practitioners tend to choose methods based on whether they are useful rather than if they are theoretically "sound" [77]. The need for rigour must therefore be balanced with the need to be practical and useable. An important pragmatic consideration that emerged as a result of empirically validating the metrics proposed was their "cost-benefit" analysis. Metrics can be very costly to collect, and will only be used if their benefits outweigh their cost of collection. Only two out of the 29 metrics originally proposed passed this test, with two new metrics discovered as a result of the action research process. This resulted in a total of four key metrics. The first two metrics really only have significance at the project level (product measures) while the second two are useful in measuring patterns of defects over time (process measures).

- *Number of entities and relationships (simplicity):* this was found to be useful for choosing between alternative models in many cases. All other things being equal, the simplest model is usually the best.
- *Development cost estimate (implementability):* this was found to be useful for making cost/quality trade-offs, choosing between alternatives and getting the database design involved in the process. This estimate was produced by the database designer.
- *Reuse percentage (integration):* this was found to be useful for calculating cost savings as a result of reuse, encouraging behavioural change and measuring improvements over time. This metric was calculated by the Information Architect.
- *Number of defects by quality factor (all factors):* this was found to be useful for identifying patterns of errors in the data modelling process and introducing preventative measures through training and process change.

The lack of success of the metrics was somewhat surprising. While all stakeholders were in favour of the concept of metrics in principle, the effort involved in calculating them quickly curbed their enthusiasm. The effort required to calculate the final set of metrics was shared between the project analyst, database designer (implementability) and Information Architect (integration), which made the workload manageable. The validation of these metrics is described in detail in [78].

#### *5.4.4. Subjective quality ratings*

In addition to the formal metrics, the review participants came up with subjective ratings for each quality factor on a scale from 1 (poor) to 5 (excellent). The analyst involved in developing the model was not required to rate the model, and different stakeholders were involved in evaluating different quality factors, as shown in Fig. 14. Each review participant was required to independently inspect the data model prior to the review and score it according to the relevant quality factors. The review session then served as a forum for reaching agreement on evaluations, recording specific quality issues and looking for ways to address defects.

A score of 4 on all quality factors was considered to be the minimum standard for acceptance (signoff) of the model. An interesting finding was that these subjective ratings were found to be more useful by project teams than the quality metrics. One reason for this is that many of the metrics defined had little comparative value—in many cases, it was difficult to tell whether a particular value was "good" or "bad".

| | Correctness | Completeness | Integrity | Flexibility | Understandability | Simplicity | Integration | Implementability |
|---|---|---|---|---|---|---|---|---|
| Business User | | ✓ | ✓ | ✓ | ✓ | | | |
| External Analyst | ✓ | | | ✓ | ✓ | ✓ | ✓ | |
| Information Architect | ✓ | | | ✓ | ✓ | ✓ | ✓ | |
| Database Designer | ✓ | | | | ✓ | | ✓ | ✓ |

Fig. 14. Stakeholder involvement in evaluating quality factors.

On the other hand, the subjective quality ratings (for example, those shown in Fig. 7) clearly define what is good and what is bad about the model, and provide clear feedback to the analyst as to where the model needs to be improved.

#### 5.4.5. *Definition of improvement strategies*

Prior to this study, specific improvement strategies had not been defined. At the beginning of this study, a set of improvement strategies was defined based on previous literature and experiences from the first action research study. Data analysts in the organisation were then asked to contribute their own ideas and experiences to this "toolkit". Whenever a quality problem was solved, analysts were encouraged to document the solution and submit it to the toolkit. The Information Architect's role was to collate contributions, classify them by quality factor and incorporate them in the toolkit. Regular meetings were held to disseminate this knowledge and to discuss new contributions. This provided a mechanism for knowledge management among DA staff, similar to "best practice databases" described in Davenport et al. [79]. As a result of this process, a comprehensive set of improvement strategies was developed. These were classified into prevention, detection and correction strategies, and are described in detail in [80].

### 5.5. *External validity*

This action research study was conducted in one of Australia's largest commercial organisations, so has high face validity. The number and diversity of projects (with budgets ranging from $50,000 to $500 million) on which the framework was used suggests that the results are generalisable to other settings.

#### 5.5.1. *Internal validity*

The following strategies were used to improve the internal validity of the results:

*Active seeking of disconfirming evidence*: the fact that the framework changed so much as a result of this study provides clear evidence that disconfirming evidence was sought and acted upon. Of the original five framework constructs, one was removed and two new constructs were added. Of the original six quality factors, two new ones were added. Finally, of the 29 metrics originally proposed, 27 were removed and two new ones added.

*Use of multiple informants*: in each action research cycle, all relevant stakeholders (business users, data analysts, database designers and the information architect) were involved in using the framework and evaluating its effectiveness. Unlike the first study, where only the authors applied the framework, in this study, members of the host organisation used it directly. This provides a much more independent test of its usefulness and applicability.

*Participation*: involving participants in the interpretation of the data as part of the observation and reflection phases can be used to strengthen interpretations of the data. The discussion between different participants and between participants and the researcher can challenge weak or inconsistent data or interpretations [81]. This corresponds to the concept of *triangulation of observers* [42]. In this study, members of the DA group were involved as participants in the research, and contributed to the refinement of the framework.

*Use of multiple cycles*: the method was applied in over 20 application development projects over a 2-year period. This also increases the generalisability of results (external validity).

*Change of practice*: the fact that the organisation incorporated the framework into their standard development practices provides objective evidence that it was perceived to be useful. By the end of this study, the framework had become an integral part of the quality management processes in the IT department.

### 5.6. *Stabilisation of the method*

Most of the changes to the framework occurred in the first four or five projects on which it was used. After this, the frequency of changes steadily reduced. While something is learned about a method almost every time it is applied, there comes a point of diminishing returns, where the effort required to conduct a further cycle outweighs the potential knowledge gained—this is where action research becomes pure action.

According to Rescher [82], the process of method development is an iterative one of

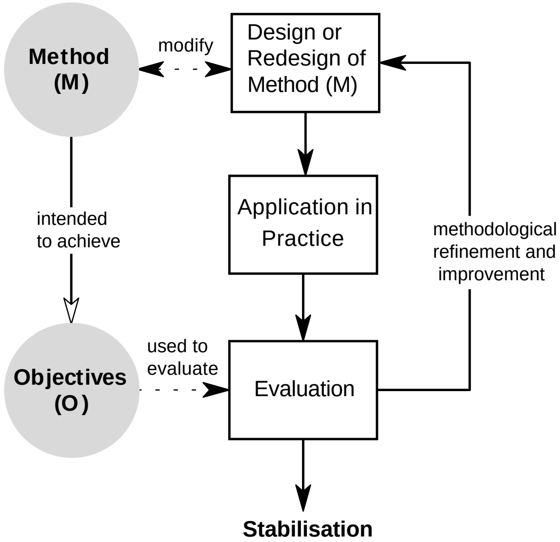


Fig. 15. Method evolution process [82].

design/redesign, application in practice, evaluation of results and improvement (Fig. 15). As one moves successively though the loops of this validation process, one arrives at a sequence of successively refined and improved methods $M_1, M_2 \ldots$ . Eventually a point of stabilisation is reached, after which all subsequent $M$ differ only trivially if at all. The fact that the framework changed so little in the latter stages of this study provides evidence that it is now a mature and stable approach.

## 6. Analysis of differences between expert and novice data modellers

This section describes how the framework was used to investigate differences in models produced by expert and novice data modellers. This study focuses on *product quality*, as the framework is used to evaluate the quality of models produced by experimental subjects.

### 6.1. Previous research

A number of laboratory based empirical studies of data modelling have compared the effectiveness of different data modelling formalisms (e.g. [83–88]). Other studies have compared expert and novice data modelling performance (e.g. [89–92]). All of these studies use the quality of the model as a basis for comparison, but define quality in terms of a limited number of quality criteria—this reflects the lack of a comprehensive quality framework for evaluating data models. Another weakness of most of these studies is that most use students rather than practitioners as subjects.

### 6.2. Method

Five of the seven quality factors from the Moody and Shanks [11] framework were used to compare expert and novice performance in an experimental study of data modelling practitioners [93]. Participants in the study were IT practitioners, who had varying levels of experience in data modelling. Each participant in the experiment developed an entity relationship model using a narrative case study transcribed from an interview with a domain expert. Each model was evaluated by three independent raters, each of whom received training in the evaluation framework. The following quality factors were used to evaluate the models:

- Correctness was evaluated in terms of the number of errors in the use of the entity relationship technique (using the instrument from Kim and March [94]).
- Completeness was evaluated in terms of number of requirements missing from the model. This was expressed as a percentage of the total user requirements (using the instrument from Kim and March [94]).
- Simplicity was evaluated in terms of the number of entities and relationships in the model (using the instrument from Moody and Shanks [11]).
- Flexibility was evaluated using a 7 point Likert scale (subjective rating).
- Understandability was evaluated using a 7 point Likert scale (subjective rating).

Finally, the overall quality of the model was evaluated using a 7 point Likert scale (subjective rating).

Table 2
Experimental results

| Quality Factor | Experts | | Novices | | Inter-rater Reliability ($\alpha$) | t-value | p-value |
|---|---|---|---|---|---|---|---|
| | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ | | | |
| Correctness | 84.35 | 17.16 | 71.66 | 17.46 | 0.97 | 2.29 | 0.029* |
| Completeness | 72.89 | 14.04 | 56.00 | 13.52 | 0.92 | 3.88 | 0.000** |
| Simplicity | 32.06 | 18.43 | 23.52 | 7.80 | 0.98 | 1.93 | 0.081 |
| Flexibility | 4.18 | 0.81 | 3.32 | 0.75 | 0.82 | 3.44 | 0.001** |
| Understandability | 3.96 | 1.03 | 3.53 | 0.77 | 0.73 | 1.50 | 0.134 |
| Overall Quality | 58.28 | 13.97 | 47.33 | 1.177 | 0.87 | 2.66 | 0.006** |

* Significant at the 0.05 level ** Significant at the 0.01 level

### 6.3. Results

Significant differences were found between models developed by experts and novices on three of the five quality factors. Table 2 summarises the results for each quality factor and overall quality.

The study found that data models designed by experts were significantly more correct, complete and flexible than models built by novices. A surprising finding was that experts produced models that were more complex than those produced by novices, although the difference was not significant ($p > 0.05$). It was expected that in the same way an expert programmer can solve a problem using fewer lines of code than a novice programmer, an expert data modeller would be able to represent the same requirements using fewer constructs [110]. However the lower average complexity of the data models produced by novices can be explained by the fact that they were missing so many requirements (completeness). Given the same level of completeness, it is expected that experts would produce simpler models.

#### 6.3.1. Interactions between quality factors

Bivariate correlation analysis was conducted to investigate interactions between all quality factors. Interactions between quality factors are important for understanding the design trade-offs implicit in the modelling process and are represented in the quality framework by the many-to-many recursive relationship on quality factor (Fig. 3). In the original formulation of the framework, expected interactions between quality factors were defined based on theory [11]. This study provided the basis for empirical analysis of these relationships. The interactions found between the quality factors are summarised in Fig. 16. Plus signs indicate positive correlations, while minus signs indicate negative correlations. The interactions are shown in both directions, as correlations are non-directional.

| | Correctness | Completeness | Simplicity | Flexibility | Understandability |
|---|---|---|---|---|---|
| Correctness | | + | | + | + |
| Completeness | + | | – | + | + |
| Simplicity | | – | | | |
| Flexibility | + | + | | | |
| Understandability | + | + | | | |

Fig. 16. Interactions between quality factors.

Strong positive correlations were found between:

- correctness and completeness ($\rho = 0.639$, $\alpha < 0.01$ * *),
- correctness and flexibility ($\rho = 0.653$, $\alpha < 0.01$ * *),
- correctness and understandability ($\rho = 0.656$, $\alpha < 0.01$ * *),
- completeness and flexibility ($\rho = 0.688$, $\alpha < 0.01$ * *),
- completeness and understandability ($\rho = 0.442$, $\alpha < 0.01$ * *).

A strong negative correlation was found between completeness and simplicity ($\rho = -0.491$, $\alpha < 0.01$ * *). This indicates that models that are more complete will also be more complex, as would be expected. It also provides some support for our supposition that the relative simplicity of novice models was primarily due to their lack of completeness.

*6.3.2. Validation of quality factors*

The experimental results were also used to evaluate the consistency, relevance and independence of the quality factors.

*Consistency (reliability)*: the high levels of inter-rater reliability on all quality factors indicate that they provide a clear and consistent basis for evaluating the quality of data models. As shown in Table 2, Cronbach alpha values for all quality factors were greater than 0.7. There is no definitive standard for reliability, but alphas of 0.7 or above are considered to be acceptable in behavioural research [95], with alphas as low as 0.5 considered acceptable in some circumstances [96].

*Relevance (validity)*: strong positive correlations were found between four of the quality factors and the overall quality of the model, which shows they are relevant determinants of quality:

- Correctness ($\rho = 0.677$, $\alpha < 0.01$ * *),
- completeness ($\rho = 0.779$, $\alpha < 0.01$ * *),
- flexibility ($\rho = 0.601$, $\alpha < 0.01$ * *),
- understandability ($\rho = 0.592$, $\alpha < 0.01$ * *).

Contrary to expectations, simplicity was found to be negatively correlated with overall quality ($\rho = 0.333$, $\alpha < 0.05$*). However, we suggested earlier that completeness was a possible confounding variable in this relationship. To test this hypothesis, a multiple regression was carried out using simplicity and completeness as independent variables and overall quality as the dependent variable. The results of this analysis showed that simplicity had a positive correlation with overall quality after controlling for the effects of completeness ($\beta = 0.252$, $\alpha < 0.05$), as predicted. This confirms the relevance of all quality factors.

A multiple regression analysis was then conducted using all quality factors as independent variables and overall quality as the dependent variable. The purpose of this was to determine the relative influence of different quality factors on perceptions of overall quality. The adjusted $r^2$ for the regression was 0.77, which means that 77% of the variation in overall quality was explained by the quality factors. This is a very high $r^2$ value, which suggests that the model is fully specified (i.e. there are no missing quality factors)—this confirms the qualitative completeness analysis conducted in the action research studies. The results for each quality factor are summarised in Table 3:

- The partial correlation coefficient ($\beta$) measures the correlation between each quality factor and perceptions of overall quality, after controlling for the effects of all other quality factors. Effectively, these represent *weightings* which are implicitly applied to each quality factor by raters in judging the overall quality of the data model.
- Percentage contribution measures the contribution of each quality factor in explaining the

Table 3
Effect of quality factors on perceptions of overall quality

| QUALITY FACTOR | PARTIAL CORRELATION ( β ) | % CONTRIBUTION |
|---|---|---|
| Understandability | 0.532 | 50.0% |
| Completeness | 0.388 | 36.4% |
| Correctness | 0.095 | 8.9% |
| Simplicity | 0.033 | 3.1% |
| Flexibility | 0.017 | 1.6% |

variance in overall quality—this is calculated as the ratio of $\beta$ values.

Surprisingly, the most influential factor was understandability, which explained 50% of the total variation in overall quality. This suggests that understandability has a much greater influence on judgements of data model quality than has been previously supposed and reflects the fact that data models are intended as a way of communicating with users. Completeness also had a strong effect on overall quality (36.4%), which would be expected. Another surprising result was that flexibility, which is widely regarded as one of the most important determinants of data model and information systems quality (e.g. [1,10,14,28,97], had an almost negligible influence on perceptions of quality.

*Independence (discriminant validity)*: *multi-collinearity* analysis was conducted as part of the multiple regression analysis to evaluate the independence of the quality factors. Tolerance values were well above 0.2 for all quality factors, which suggests that they are all independent (though related) determinants of overall quality.

### 6.4. Theoretical and practical significance

#### 6.4.1. Theoretical significance

The framework provided the basis for reliable evaluation and a more comprehensive view of the differences between expert and novice models than previous experimental studies. The framework may provide a useful instrument in conducting experimental studies of data modelling in the future. The experiment also provided useful information about the interactions between quality factors, and the relative influence of different quality factors on judgements of overall quality.

#### 6.4.2. Practical significance

Understanding the nature of differences between models produced by experts and novices provides information which can be used to improve the teaching of data modelling. Use of practitioners rather than students increased the generalisability of the results to practice.

## 7. Automated tool development

### 7.1. Systems development as a research method

Systems development is a research method in which scientific knowledge is used to produce devices, systems or methods including design and development of prototypes [98]. In this approach, theory is used to develop a prototype system, which is then used to test the theory. It thus provides a way of linking basic and applied research [99]. According to Nunamaker et al. [37]:

> The development of a method or system can provide a perfectly acceptable piece of evidence (an artifact) in support of a 'proof', where proof is taken to be any convincing argument in support of a worthwhile hypothesis. Systems development could be thought of as a 'proof by demonstration'.

This section describes how the systems development approach was used to develop an automated tool based on the data model evaluation framework. A laboratory experiment was then conducted to evaluate the usefulness of the tool in supporting the task of evaluation.

### 7.2. The data model quality advisor

The data model quality framework was incorporated into an automated tool called the Data Model Quality Advisor (DMQA) [100]. The DMQA acts as an "expert assistant" in the evaluation of data models. It provides a hypertext explanation facility for the constructs of the quality evaluation framework, and supports evaluation and comparison of up to three data models at a time.

The hypertext explanation facility provides a graphical view of the framework as a user interface. Explanations and examples of any of the framework constructs can be viewed by selecting the appropriate icon within the graphical model. The user is able to navigate amongst constructs of the framework using the hypertext links provided. Users can also access the framework via the meta-model of the quality evaluation framework (shown earlier in Fig. 3) as an alternative user interface.

The evaluation and comparison facility of the DMQA supports the allocation of weightings to each quality factor by different stakeholders. Ratings for each quality factor for up to three alternative data models may then be entered by each stakeholder. These are stored for subsequent comparison. The DMQA also provides advice on alternative ways of evaluating the quality factors. After all stakeholders' ratings have been entered, a summary of their evaluations with rankings for the alternative models can be displayed. The user may seek explanation of any component of the framework during evaluation and comparison using the explanation facility. Fig. 17 shows the ratings summary screen of the DMQA.

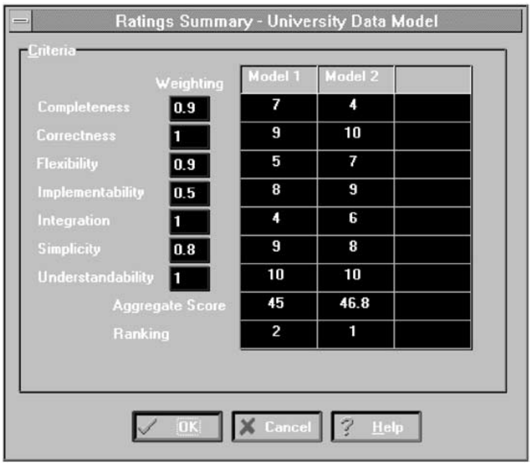


Fig. 17. Ratings summary screen of the DMQA.

### 7.3. Experimental evaluation

A laboratory experiment was conducted to examine the usefulness and usability of the quality evaluation framework and the DMQA in evaluating data models. A one group, post-test only design was used—this is called a one shot case study [101]. The study involved 20 experienced data modelling practitioners and academics. The participants in the study were each asked to use the DMQA to learn about the quality evaluation framework and then to evaluate three alternative data models for a small case example.

Each participant then completed a questionnaire about the evaluation framework the DMQA. Participants were asked to indicate the extent to which they agreed or disagreed with each statement using a 5-point Likert scale (1 "strongly agree" to 5 "strongly disagree"). The results obtained in the study are summarised in Table 4. The table shows the mean and standard deviation for ratings for each statement together with the results of a *t*-test to determine if the average rating was significantly different to three (the mid-point of the scale). This was used to determine whether responses were significantly positive or negative.

The empirical study indicated strong support for the need to evaluate the quality of data models and

Table 4
Summary of questionnaire statements and participant responses

| Statement | Mean Response | t-value | Significance Level | Result |
|---|---|---|---|---|
| 1. Evaluating the quality of a conceptual data model is critical to the successful development of an information system | 1.55 (0.60) | -10.808 | $\alpha < 0.01$** | YES |
| 2. A framework for evaluating the quality of conceptual data models would constrain the way practitioners prefer to work | 3.15 (0.81) | 0.828 | Not significant | Undecided |
| 3. The tool is useful when evaluating the quality of data models | 2.25 (0.78) | -4.300 | $\alpha < 0.01$** | YES |
| 4. The tool is useful in comparing alternative data models | 1.80 (0.83) | -6.466 | $\alpha < 0.01$** | YES |
| 5. The tool is irrelevant to understanding and using the quality framework | 3.55 (.083) | -2.963 | $\alpha < 0.05$* | NO |

for the usefulness of the DMQA in understanding the quality evaluation framework and in evaluating and comparing data models. However participants were unsure whether use of the framework would constrain the way practitioners worked. Surprisingly, the second action research study showed that use of the framework actually encouraged innovation rather than constraining people.

### 7.4. Strengths and weaknesses of the research

The results of this study should be interpreted with caution, as this was an exploratory study, with inherent methodological weaknesses. More rigorous empirical testing of the DQMA is planned as part of further research.

*Internal validity*: the one-shot case study is the weakest possible experimental design in terms of internal validity [101]. The lack of a pre-test and control group make it inadequate for establishing causality [102]. In this case, there is no comparison between participants using the tool and participants not using the tool, so it is impossible to draw strong conclusions about whether the tool improves task performance.

External validity: although the study was conducted using experienced practitioners, it was also conducted in a laboratory setting, which limits its generalisability to practice. The artificiality of the laboratory is one of the major weaknesses of the experimental method [103]. As Further testing of the tool is planned in a real world setting.

## 8. Conclusion

This paper has described how the data model quality evaluation framework proposed by Moody and Shanks [11] has been validated using a variety of research methods. Experiences in practice have been used to refine the framework using an action research approach. The paper describes how the framework has been used to:

(a) Quality assure individual data models as part of application development projects (product quality),
(b) reengineer application development procedures to build quality into the data modelling process (process quality),
(c) investigate differences between data models produced by expert and novice data modellers,
(d) build an expert assistant to support the evaluation process (the Data Model Quality Advisor).

The framework has been successfully used in a wide range of project environments, and has evolved to the point where it is now relatively stable and complete.

### 8.1. Practical significance (contributions to improved data modelling practices)

The major practical contribution of this paper is that it shows how the data model quality framework can be applied in practice to improve the quality of data models. Because of the critical role data modelling plays in systems development, even small improvements in the quality of data models are likely to have a significant impact on the quality of the final system. Use of the framework led to significant practical benefits for the organisations in which it was applied, in terms of development cost savings and quality improvements.

Improving the quality of data models provides a major opportunity for organisations to improve the productivity of systems development. Empirical studies show that more than half the errors which occur during systems development are the result of inaccurate or incomplete requirements [104,105]. Also, the most common reason for the failure of systems development projects is incomplete requirements [106,107]. The action research studies showed that use of the framework reduced requirements errors by almost 30%. Finding and correcting these errors during analysis has the potential to reduce error correction costs by a factor of more than 100 compared to later on in the lifecycle [8].

In both action research studies, the most significant benefits were achieved through integration of data models. In the second action research study, data reuse levels more than doubled as a result of using the framework, leading to estimated savings of 42.8% in development costs. Data reuse also helps to prevent ongoing costs of data duplication, which include re-keying of data, storage costs, synchronisation and reconciliation costs [4].

### 8.2. Theoretical significance (contributions to theoretical understanding of data modelling)

This paper addresses the two major deficiencies in the existing research on data model quality identified in Section 1.

#### 8.2.1. Empirical validation

The framework has been extensively tested and refined in practice as part of an ongoing action research programme. The framework was modified extensively as a result of the research. This is the first time a data model quality framework has been validated in an organisational context using real practitioners. The range of project environments in which the framework has been applied (more than 20 projects in two large commercial organisations) suggests that the results are generalisable to other settings. The framework has also been validated in laboratory settings.

#### 8.2.2. Process quality

Like all of the frameworks previously proposed, this framework was originally focused exclusively on product quality (evaluation of a completed data model). As a result of the action research programme, the framework was extended to include process quality aspects (how to develop a data model in a high quality manner). A major finding of the research was that the most significant benefits are achieved through improving the process of data modelling (error prevention) rather than through quality assuring the final result (error detection and correction). This is consistent with the findings of the TQM literature, which maintains that sustainable improvements in quality can only be achieved by modifying production processes to prevent defects [12]. Previous research in this area has ignored issues of product quality.

#### 8.2.3. Other theoretical contributions

Other theoretical contributions of the research include:

*Expert vs. novice data modelling performance*: the framework provided the basis for more comprehensive understanding of the differences in the quality of models produced by experts and novices. It also provided the basis for the development of reliable instruments for evaluating these differences.

*"Soft" vs. "hard" information*: the research provided an insight into the relative importance of "soft" and "hard" information in data model quality management. Contrary to expectations of both the researchers and the research participants, subjective quality ratings and qualitative descriptions of quality issues were found to be more useful than quantitative measures in the quality improvement process. Metrics had limited applicability, which may reflect the qualitative nature of analysis compared to design: in database design quantitative data (performance and storage space estimates) play a much more important role. This suggests that attempts to quantify data model quality may be counterproductive in practice.

*People issues*: the research provided an understanding of the importance of people issues in the quality management process. In the action research studies, the lack of involvement of particular stakeholders and differences in stakeholder perspectives was found to be at the heart of most quality problems. Involvement of all stakeholders in the data modelling process was found to be more important than any other single issue in achieving quality improvements.

#### 8.2.4. Multi-methodological validation of IS design methods

Validation of IS design methods has been a vexed issue in the IS field (e.g. [19,21,24–28]. There are fundamental problems validating "knowledge how" (methods) as compared to "knowledge that" (theses), which are the normal domain of scientific research [20,108]. This paper shows how multiple research methods, field and laboratory based, qualitative and quantitative, can be used in combination to validate a design method. The use of multiple research methods in this way fits

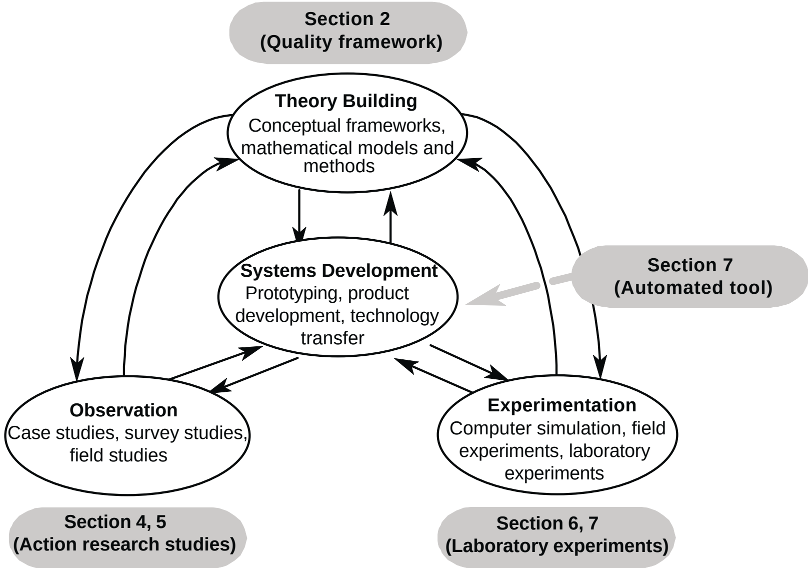


Fig. 18. A multi-methodological approach to IS research (adapted from [37]).

into Nunamaker et al. [37] multi-methodological approach to IS research (Fig. 18).

Action research proved to be a valuable technique for bridging the gap between research and practice, and may provide a generally useful approach to validating IS design methods [43]. Its major advantage is that it provides a means for "validating methods in an organisational context using real practitioners", which is something which is sorely needed in IS design research [21].

## References

[1] G.C. Witt, G.C. Simsion, Data Modeling Essentials: Analysis, Design, and Innovation, The Coriolis Group, 2000.

[2] Asma, ASMA Project Database Release 7.0, ASMA (Australian Software Metrics Association), P.O. Box 1287, Box Hill, Victoria, Australia, 3128, November, 1996.

[3] Gartner Research, Sometimes You Gotta Break the Rules. Gartner Group Strategic Management Series Key Issues, November 23, 1992.

[4] D.L. Moody, G.C. Simsion, Justifying investment in information resource management, Aust. J. Inform. Systems 3 (1) (1995) 25–37.

[5] R.D. Banker, R.J. Kauffman, Reuse and productivity in integrated computer aided software engineering: an empirical study, MIS Q. 15 (3) (1991) 375–401.

[6] J.C. van Vliet, Software Engineering: Principles and Practice, Wiley, Chichester, England, 1993.

[7] R.E. Zultner, The Deming way: total quality management for software, Proceedings of Total Quality Management for Software Conference, Washington, DC, April, 1992.

[8] B.W. Boehm, Software Engineering Economics, Prentice-Hall Inc., Englewood Cliffs, NJ, 1981.

[9] C. Walrad, E. Moss, Measurement: the key to application development quality, IBM Systems J. 32 (3) (1993) 445–460.

[10] J. Krogstie, O.I. Lindland, G. Sindre, Towards a deeper understanding of quality in requirements engineering, Proceedings of the Seventh International Conference on Advanced Information Systems Engineering (CAISE), Jyvaskyla, Finland, June, 1995.

[11] D.L. Moody, G.G. Shanks, What makes a good data model? Evaluating the quality of entity relationship models, in: P. Loucopolous, (Ed.), Proceedings of the 13th International Conference on the Entity Relationship Approach, Manchester, England, December 14–17, 1994.

[12] J.R. Evans, W.M. Lindsay, The Management and Control of Quality, 5th Edition, South-Western (Thomson Learning), Cincinnati, OH, 2002.

[13] W.E. Deming, Out of the Crisis, MIT Center for Advanced Engineering, Cambridge, MA, 1986.

[14] B. von Halle, Data: asset or liability? Database Programming Design 4 (7) (1991) 21–24.

[15] C. Batini, S. Ceri, S.B. Navathe, Conceptual Database Design: An Entity Relationship Approach, Benjamin Cummings, Redwood City, CA, 1992.
[16] A. Levitin, T. Redman, Quality dimensions of a conceptual view, Inform. Process. Manage. 31 (1) (1994).
[17] O.I. Lindland, G. Sindre, A. Sølvberg, Understanding quality in conceptual modelling, IEEE Software 11 (2) (1994) 42–49.
[18] S. Kesh, Evaluating the quality of entity relationship models. Information Software Technol. 37 (12) (1995) 681–689.
[19] J. Ivari, Dimensions of information systems design: a framework for a long range research program, Inform. Systems J. 11 (4) (1986) 343–355.
[20] N. Rescher, Methodological Pragmatism: Systems-Theoretic Approach to the Theory of Knowledge, Basil Blackwell, Oxford, 1977.
[21] J.L. Wynekoop, N.L. Russo, Studying systems development methodologies: an examination of research methods, Inform. Systems J. 7 (1) (1997) 47–66.
[22] M. Gibbons, C. Limoges, H. Nowotny, S. Schwartzman, P. Scott, M. Trow, The New Production of Knowledge: The Dynamics of Science and Research in Contemporary Societies, Sage Publications, Beverley Hills, CA, 1994.
[23] D.L. Moody, Metrics for evaluating the quality of entity relationship models, in: T.W. Ling, S. Ram, M.L. Lee (Eds.), Proceedings of the International Conference on Conceptual Modelling (ER '98), Elsevier, Singapore, Lecture Notes in Computer Science, Springer, Berlin, November 16–19, 1998.
[24] T.W. Olle, H.G. Sol, A.A. Verrijn-Stuart (Eds.), Information Systems Design Methodologies: A Comparative Review, North-Holland, Amsterdam, 1982.
[25] T.W. Olle, H.G. Sol, C.J. Tully (Eds.), Information Systems Design Methodologies: A Feature Analysis, North-Holland, Amsterdam, 1983.
[26] T.W. Olle, H.G. Sol, A.A. Verrijn-Stuart (Eds.), Information Systems Design Methodologies: Improving the Practice, North-Holland, Amsterdam, 1986.
[27] G. Fitzgerald, in: H.E. Nissen, H.K. Klein, R. Hirschheim (Eds.), Validating New Information Systems Techniques: A Retrospective Analysis, Information Systems Research: Contemporary Approaches And Emergent Traditions, North-Holland, Amsterdam, 1991.
[28] R.A. Weber, Ontological Foundations of Information Systems, Coopers and Lybrand Accounting Research Methodology Monograph No. 4, Coopers and Lybrand, Melbourne, Australia, 1997.
[29] J.A. Bubenko, in: T.W. Olle, H.G. Sol, A.A. Verrijn-Stuart (Eds.), Information Systems Methodologies—A Research View, Information Systems Design Methodologies: Improving The Practice, North-Holland, Amsterdam, 1986.
[30] B. Curtis, in: E. Soloway, S. Iyengar, (Eds.), By The Way, Did Anyone Study Any Real Programmers? Empirical Studies of Programmers, Norward, NJ, Ablex, 1986.
[31] D.L. Moody, G.G. Shanks, Evaluating and improving the quality of entity relationship models: an action research programme, Aust. Comput. J. 30 (3) (1998) 97–110.
[32] C. Westrup, Information systems methodologies in use, J. Inform. Technol. 8 (1993) 267–275.
[33] N. Rescher, Cognitive Systematization, Basil Blackwell, Oxford, 1979.
[34] R.L. Baskerville, T. Wood-Harper, A critical perspective on action research as a method for information systems research, J. Inform. Technol. 3 (11) (1996) 235–246.
[35] R.D. Galliers, in: H.E. Nissen, H.K. Klein, R. Hirschheim, (Eds.), Choosing Information Systems Research Approaches, Information Systems Research: Contemporary Approaches and Emergent Traditions, North-Holland, Amsterdam, 1991.
[36] R.D. Galliers, Information Systems Research: Issues, Methods and Practical Guidelines, Blackwell Scientific Publications, Oxford, 1992.
[37] J. Nunamaker, M. Chen, T.D.M. Purdin, Systems development in information systems research, J. Manage. Inform. Systems 7 (3) (1991) 89–106.
[38] G. Shanks, A. Rouse, D. Arnott, A review of approaches to research and scholarship in information systems, Proceedings of the Fourth Australian Conference on Information Systems, Brisbane, 1993.
[39] T.D. Jick, Mixing qualitative and quantitative methods: triangulation in action, Administrative Sci. Q. 24 (1979) 602–611.
[40] B. Kaplan, D. Duchon, Combining qualitative and quantitative methods in information systems research: a case study, MIS Q. 12 (4) (1988) 571–586.
[41] A. Lee, Integrating positivist and interpretivist approaches to organisational research, Organ. Sci. 2 (4) (1991) 342–365.
[42] W.L. Neuman, Social Research Methods—Qualitative and Quantitative Approaches, 4th Edition, Allyn and Bacon, Needham Heights, MA, 2000.
[43] D. Avison, F. Lau, M. Myers, P.A. Nielsen, Action Research, Comm. ACM 42 (1) (1999) 94–97.
[44] T.L. Baker, Doing Social Research, McGraw-Hill, New York, 1998.
[45] J. Mckernan, Curriculum Action Research: A Handbook of Methods and Resources for the Reflective Practitioner, Kogan Page, London, 1991.
[46] E.T. Stringer, Action Research—A Handbook for Practitioners, SAGE publications, London, 1996.
[47] J. Masters, The history of action research, in: I. Hughes (Ed.), Action Research Electronic Reader (on-line), http://www.behs.cchs.usyd.edu.au/arow/Reader/rmasters.htm, The University of Sydney, 1995.
[48] B. Dick, A beginner's guide to action research [On line], in: B. Dick, R. Passfield, P. Wildman, (Eds.), Action Research and Evaluation On Line (AREOL), Available on-line at http:\\\www.scu.edu.au/schools/gcm/ar/arp/guide.html, 2000.

[49] P.B. Checkland, From framework through experience to learning: the essential nature of action research, in: H.E. Nissen, H.K. Klein, R. Hirschheim, (Eds.), Information Systems Research: Contemporary Approaches And Emergent Traditions, North-Holland, Amsterdam, 1991.

[50] D.L. Moody, Building links between IS research and professional practice: improving the relevance and impact of IS research, in: R.A. Weber, B. Glasson, (Eds.), International Conference on Information Systems (ICIS'00), Brisbane, Australia, December 11–13, 2000.

[51] F.A. Lau, A review of action research in information systems studies, in: A. Lee, J. Liebenau, J.I. DeGross, (Eds.), Information Systems and Qualitative Research, 1997, pp. 31–68.

[52] M. Oosthuizen, Action research, in: K. Williamson, (Ed.), Research Methods for Students and Professionals: Information Management and Systems, Charles Sturt University, Centre for Information Studies, 2000.

[53] K. Lewin, Action research and minority problems, J. Social Issues 2 (1946) 34–46.

[54] S. Kemmis, R.E. McTaggart, The Action Research Planner, Deakin University, Melbourne, Australia, 1988.

[55] I.M. Holter, D. Schwartz-Barcott, Action research: what is it, how has it been used and how can it be used in nursing? J. Adv. Nurs. 18 (1993) 296–304.

[56] R. Hatten, D. Knapp, R. Salonga, Action research: comparison with the concepts of the reflective practitioner and quality assurance, in: I. Hughes, (Ed.), Action Research Electronic Reader (on-line), http://www.behs.cchs.usyd.edu.au/arow/Reader/rmasters.htm, The University of Sydney, 1997.

[57] I. Benbasat, R.W. Zmud, Empirical research in information systems: the practice of relevance, MIS Q. 23 (1) (1999) 3–16.

[58] T.H. Davenport, M.L. Markus, Rigour vs. relevance revisited: response to benbasat and zmud, MIS Q. 23 (1) (1999) 18–23.

[59] R.D. Galliers, Relevance and rigour in information systems research: some personal reflections on issues facing the information systems research community, Proceedings of the IFIP TC8 Conference on Business Process Reengineering: Information Systems and Challenges, Gold Coast, Australia, 1994.

[60] P.G.W. Keen, Relevance and rigour in information systems research: improving quality, confidence, cohesion and impact, in: H.-E. Nissen, H.K. Klein, R. Hirschheim, (Eds.), Information Systems Research: Contemporary Approaches and Emergent Traditions, Elsevier Science Publishers, North-Holland, Amsterdam, 1991.

[61] P.B. Checkland, S. Holwell, Information, Systems and Information Systems: Making Sense of the Field, Wiley, New York, 1998.

[62] S.L. Chow, Significance test or effect size, Psychol. Bull. 103 (1) (1988) 105–110.

[63] B. Dick, Rigour and relevance in action research [On line], in: B. Dick, R. Passfield, P. Wildman, (Eds.), Action Research, Evaluation On Line (AREOL), Available on-line at http:\\\www.scu.edu.au/schools/gcm/ar/arp/guide.html, 1997.

[64] C. Alexander, Notes on The Synthesis of Form, Harvard University Press, Boston, 1968.

[65] Y.-G. Kim, G.C. Everest, Building an IS Architecture: collective wisdom from the field, Inform. Manage. 26 (1995) 1–10.

[66] B. Dick, Approaching an action research thesis: an overview, in: B. Dick, R. Passfield, P. Wildman, (Eds.), Action Research and Evaluation on Line (AREOL), Available on-line at http:\\\www.scu.edu.au/schools/gcm/ar/arp/guide.html, 1997.

[67] C. Argyris, R. Putnam, D.M. Smith, Action Science: Concepts, Methods and Skills for Research and Intervention, Jossey-Bass, San Francisco, 1985.

[68] T.H. Davenport, Saving IT's soul: human centred information management, Harvard Business Review, March–April, 1994.

[69] R.E. Zultner, QFD for software: satisfying customers, American Programmer, February, 1992, pp. 1–14.

[70] G.C. Simsion, Creative data modelling, Proceedings of the Tenth International Entity Relationship Conference, San Francisco, 1991.

[71] T.P. Moran, J.P. Carroll (Eds.), Design Rationale: Concepts, Techniques and Use, Lawrence Erlbaum Associates, New Jersey, 1996.

[72] R.G. Ross, The Business Rule Book, 2nd Edition, Database Research Group, Boston, MA, 1996.

[73] B. von Halle, J. Conkey, Rewiring the business, Database Programming Design 10 (1) (1997) 11–15.

[74] ISO, Information processing systems: concepts and terminology for the conceptual schema and the information base, Information Standards Organisation (ISO), ISO Technical Report 9007, 1987.

[75] A.D. Baddeley, The magical number seven: still magic after all these years? Psychol. Rev. 101 (2) (1994) 353–356.

[76] G.A. Miller, The magical number seven, plus or minus two: some limits on our capacity for processing information, Psychol. Rev. 63 (1956) 81–97.

[77] W. Kent, Data and Reality: Basic Assumptions in Data Processing Reconsidered, North-Holland, Amsterdam, NY, 1978.

[78] D.L. Moody, Metrics for improving the quality of entity relationship models: an empirical investigation, University of Melbourne Information Systems Working Paper, Department of Information Systems, University of Melbourne, January (under submission), 2000.

[79] T.H. Davenport, D.W. de Long, M.C. Beers, Successful knowledge management projects, Sloan Manage. Rev. 39 (2) (1998) 43–52.

[80] D.L. Moody, Strategies for improving the quality of entity relationship models, Information Resource Management Association (IRMA) Conference, Anchorage, Alaska, Idea Group Publishing, May 21–24, 2000.

[81] B. Dick, Sources of rigour in action research: addressing the issues of trustworthiness and credibility, Association

for Qualitative Research Conference "Issues of Rigour in Qualitative Research", Duxton Hotel, Melbourne, Victoria, 6–10 July 1999.

[82] N. Rescher, The Primacy of Practice, Basil Blackwell, Oxford, 1973.

[83] D. Batra, J.A. Hoffer, R.P. Bostrom, Comparing representations with relational and EER models, Commun. ACM 33 (2) (1990) 126–139.

[84] D.B. Bock, T. Ryan, Accuracy in modelling with extended entity relationship and object oriented data model, J. Database Manage. 4 (4) (1993) 30–39.

[85] B.C. Hardgrave, N.P. Dalal, Comparing object oriented and extended entity relationship models, J. Database Manage. 6 (3) (1995) 15–21.

[86] H. Lee, B.G. Choi, A comparative study of conceptual data modelling techniques, J. Database Manage. 9 (2) (1998) 26–35.

[87] P. Shoval, M. Even-Chaime, Database schema design: an experimental comparison between normalisation and information analysis, Database 18 (3) (1987) 30–39.

[88] P. Shoval, S. Shiran, Entity-relationship and object-oriented data modeling: an experimental comparison of design quality, Data Knowledge Eng. 21 (1997) 297–315.

[89] D. Batra, J. Davis, Conceptual data modelling in database design: similarities and differences between expert and novice designers, Int. J. Man-Mach. Stud. 37 (1992) 83–101.

[90] P. Chaiyasut, G.G. Shanks, Conceptual data modelling process: a study of novice and expert data modellers, in: T. Halpin, R. Meersman, (Eds.), Proceedings of The First International Conference on Object Role Modelling, Magnetic Island, Queensland, Australia, July, 1994.

[91] N. Maiden, A. Sutcliffe, Analysing the novice analyst: cognitive models in software engineering, Int. J. Man Mach. Stud. 367 (1992) 719–740.

[92] G.G. Shanks, G.C. Simsion, M. Rembach, The role of experience in conceptual data modelling, Fourth Australian Information Systems Conference, Brisbane, Australia, 1993.

[93] G.G. Shanks, Conceptual data modelling: an empirical study of expert and novice data modellers, Aust. J. Inform. Systems 4 (2) (1997) 63–73.

[94] Y.-G. Kim, S.T. March, Comparing data modelling formalisms, Commun. ACM 38 (1995) 6.

[95] J. Nunally, Psychometric Theory, 2nd Edition, McGraw-Hill, New York, 1978.

[96] R.D. Caplan, R.K. Naidu, R.C. Tripathi, Coping and defense: constellations vs. components, J. Health Social Behav. 25 (1984) 303–320.

[97] H.A. Simon, Sciences of The Artificial, 3rd Edition, MIT Press, Cambridge, MA, 1996.

[98] J. Nunamaker, M. Chen, Systems Development in Information Systems Research, IEEE631-639, 1990.

[99] F. Burstein, S. Gregor, The systems development or engineering approach to research in information systems: an action research perspective, Proceedings of the Tenth Australasian Conference on Information Systems, Adelaide, Australia, December 3–5, 1999.

[100] G. Shanks, P. Darke, Quality in conceptual modelling: linking theory and practice, Proceedings of the Pacific Asia Conference on Information Systems (PACIS), Queensland University of Technology, Brisbane, Australia, May, 1997.

[101] D.T. Campbell, J.C. Stanley, Experimental and Quasi-Experimental Designs for Research, Houghton Mifflin College, 1966.

[102] D.R. Cooper, P.S. Schindler, Business Research Methods, 6th Edition, McGraw-Hill International, Singapore, 1998.

[103] E.R. Babbie, The Practice of Social Research, Wadsworth Publishing, Belmont, CA, 1998.

[104] S. Lauesen, O. Vinter, Preventing requirement defects, Proceedings of the Sixth International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ'2000), Stockholm, Sweden, June 5–6 2000.

[105] J. Martin, Information Engineering, Prentice-Hall, Englewood Cliffs, NJ, 1989.

[106] Standish group, The CHAOS Report into Project Failure, The Standish Group International Inc. Available on-line at http://www.standishgroup.com/visitor/chaos.htm, 1995.

[107] Standish group, Unfinished Voyages, The Standish Group International Inc. available on-line at http://www.standishgroup.com/visitor/voyages.htm, 1996.

[108] D.L. Moody, Dealing with complexity: a practical method for representing large entity relationship models, Ph.D. Thesis, Department of Information Systems, University of Melbourne, Melbourne, Australia, 2001.

[109] E. Mumford, Effective Systems Design and Requirement Analysis. The ETHICS Approach, Macmillan Press, Basingstoke, UK.

[110] D.L. Moody, The seven habits of highly effective data modellers, Database Programming and Design October (1996) 12–30.