



IE\Q: A methodology for implementing a quality system for information management

R. Veryard

*TI Information Engineering Ltd,
Sunbury-upon-Thames, UK*

ABSTRACT

This paper describes IE\Q, which is a methodology for designing, implementing and operating a quality system for the Information Management function, primarily targeted at organizations and projects using the Information Engineering methodology.

IE\Q operates at two levels: for the Information Management function as a whole, and for individual IS development projects. These two levels are fully integrated, allowing for both top-down and bottom-up implementations of the quality system.

When applied to the Information Management function as a whole, IE\Q leads to a quality system that is fully compliant with ISO 9000, and with TickIT. IE\Q can therefore be used as a methodology for attaining certification against either or both of these standards. IE\Q also contains guidelines covering the scope of the SEI Capability Maturity Model.

The whole area of quality systems methodologies is a new one, and IE\Q is therefore an important rationalizing step of quality management.

INTRODUCTION

During the 1980s, methodologies for IS development reached maturity. The topic has been examined by many conferences (notably the CRIS series) and textbooks, and added to most university syllabuses. Automation has progressed from simple modelling tools and data dictionaries (in 1980) to integrated CASE tools (by 1990). It is now commonplace, at least for large IS



36 Software Quality Management

organizations, to have adopted a standard methodology such as Information Engineering or SSADM for the IS development process, and to use CASE tools to support this process. We may of course expect further progress during the 1990s - methods will converge (through such initiatives as Euromethod) and new techniques and targets will be introduced (such as distribution and object-orientation) - but progress will be incremental rather than revolutionary.

In the 1990s, the focus of attention now shifts to the management level, where such issues as risk management, technology assimilation and quality need to be systematically addressed. The IS manager faces a complex set of demands; there is an increasing quantity of material in the public domain that addresses these demands, but in a fragmented way. Management-level methodologies are needed to guide the IS manager through these issues and demands in a structured and effective way; these methodologies will complement the existing engineering-level methodologies, and enhance the overall capability/maturity of the IS organization.

This paper describes a methodology called IE\Q that we have developed within TI Information Engineering (formerly JMA) to steer the IS manager through the minefield of quality. It is based on Information Engineering, but can be easily adapted to other methodologies such as SSADM.

ON DESCRIBING A METHODOLOGY

At an abstract level, a methodology is a system to transform a set of inputs into a set of outputs. To describe such a system fully, we need to know:

- 1 What is the outcome (or set of possible outcomes) that the methodology can deliver?
- 2 What are the possible starting conditions for the methodology?
- 3 How do we get from the starting conditions to the desired outcome?
- 4 What are the costs of getting from the starting conditions to the desired outcome?
- 5 What are the risks of not getting to the desired outcome at all? What are the risks of getting a poorer outcome slower, or incurring a higher cost?



If we want to evaluate such a system, it is useful to know the empirical and conceptual basis for the methodology.

- 6 What empirical evidence is there that the methodology works?
- 7 What rational / conceptual / logical arguments are there that the methodology works?

If we want to use such a system, it is also useful to know the support provided for the methodology.

- 8 What tools and techniques are available for use with the methodology? What skills are required, and how easily are these skills acquired?
- 9 What management disciplines are required by the methodology? Does the methodology itself offer these disciplines?
- 10 How easy will it be to implement this methodology into my environment?

IE\Q GOALS

If you want to do something systematic about quality, then by definition your goal is to have a quality system. Thus the goal of IE\Q is to implement a quality system into an information management organization.

The standard structure and requirements for such a quality system are given by the ISO 9000 family of standards, and by various interpretation documents such as the TickIT Guide. IE\Q takes this to be the template for a quality system. If ISO 9000 or TickIT certification is required - and it often is - then you will need to stick fairly closely to this template. If certification is not required, then the organization may choose to deviate from the template, but any such deviations need to be carefully argued. Even if certification is required, this may not be practical in the short term: for some organizations it may take two or three planning cycles before the quality system is complete and certifiable. IE\Q therefore provides a framework for deciding what elements of the quality system are required immediately, and what elements are required in the medium term.

What is a quality system worth to an organization? There are at least four ways in which a quality system can pay back the investment:

- by improving quality of product or service



38 Software Quality Management

- by improving quality of production
- by reducing risk
- by improving external confidence in products and services

Although it is difficult to put a specific monetary value on these benefits, it is these benefits (or some combination of them) that justifies the effort.

IE\Q APPROACH

Any implementation methodology must set its pace according to the culture and resources of the organization. IE\Q therefore adopts a planning cycle approach, which identifies the quality drivers and concentrates effort on the priority areas.

One of the trends of the 1990s is increasing distribution, both of the information systems and technologies themselves, and of the organizations that develop and manage them. From a quality management point of view, this trend can be encouraged, as long as it brings the information service closer to the end-user without compromising quality. IE\Q therefore models the distribution of responsibilities to support a distributed environment, thus obviating a large central quality staff.

IE\Q LEVELS

Quality can be managed at many levels.

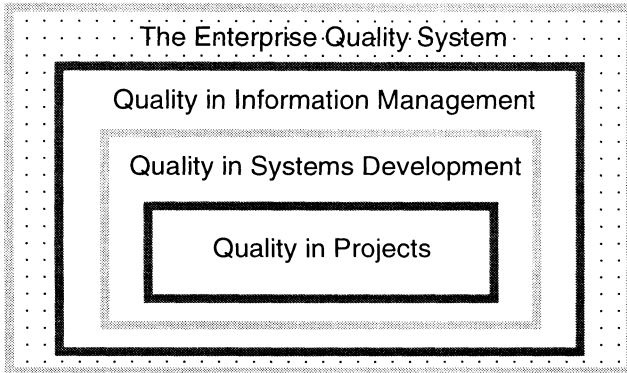


Figure 1 - Layers of quality system

Ideally the quality system applies to the people and operations of the whole enterprise. However, it is perfectly possible to implement a quality system dealing with exactly the same range of topics even down at the level of a single project.

The quality system must conform to the policies and objectives of the enterprise as a whole. IE\Q develops Information Management quality policies and plans from the enterprise quality policy.

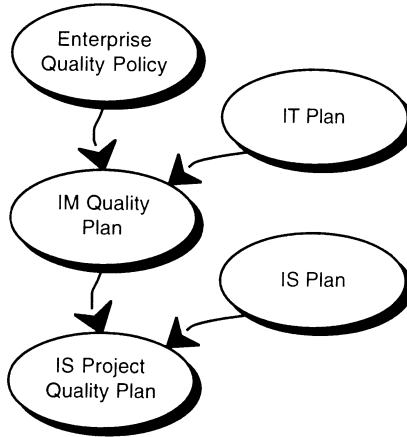


Figure 2 - Top-down derivation of policies and plans

From an analysis of the business intentions of the main agents or stakeholders, we can specify the goals of the quality system. This specification is encapsulated in a quality plan, together with a series of quality policies, aimed at achieving or maintaining defined levels of quality for the products and processes of information management.

The quality system itself will develop these quality intentions further, clarifying the plans and detailing the policies.

IE\Q TASK STRUCTURE

IE\Q consists of three main tasks, as shown in the diagram. This task structure applies, with some variations, at the organizational level and at the project level.



40 Software Quality Management

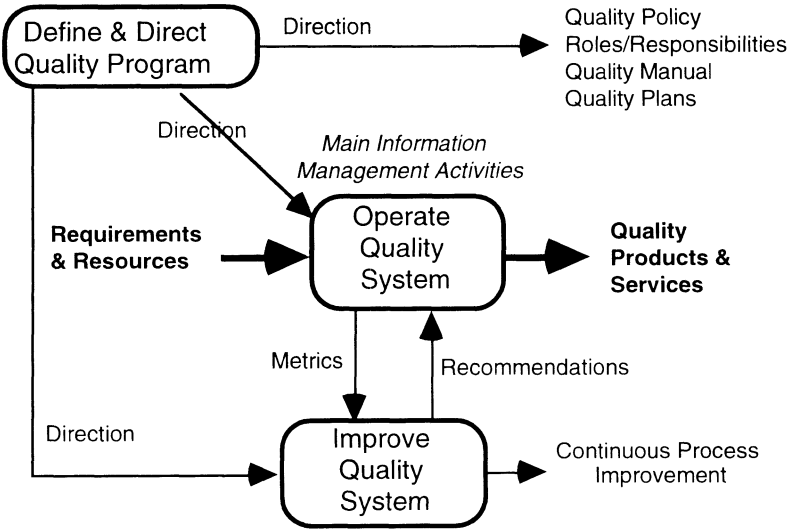


Figure 3 - Overall task structure

Now let us look at each of these three tasks in more detail.

Define and Direct Quality Program

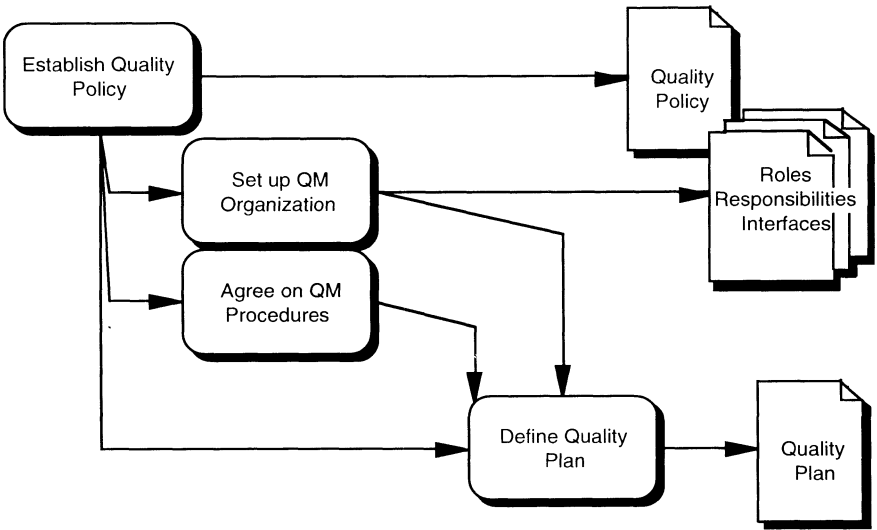


Figure 4 - Task Structure: Plan Quality System

The key features of the planning stage are the formulation and agreement of policies, responsibilities and plans for the quality system as a whole, and for individual projects.

Operate Quality System

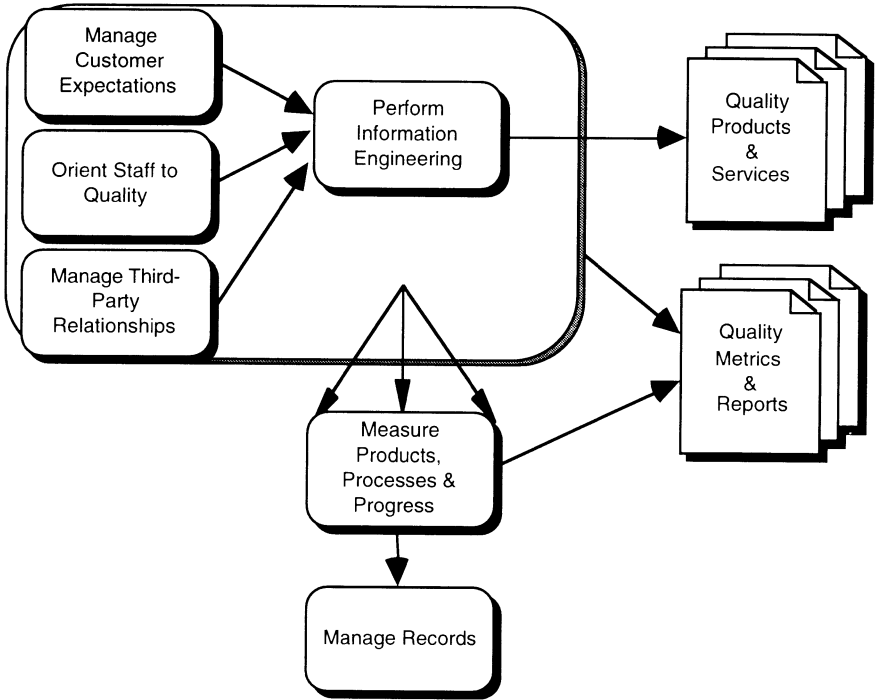


Figure 5 - Task structure: Operate Quality System

The key feature of the operational stage of the quality system is the implementation of full quality management, by establishing good communications and cooperative interworking for your staff, with one another, with customers, and with third-party suppliers.

42 Software Quality Management

Improve Quality System

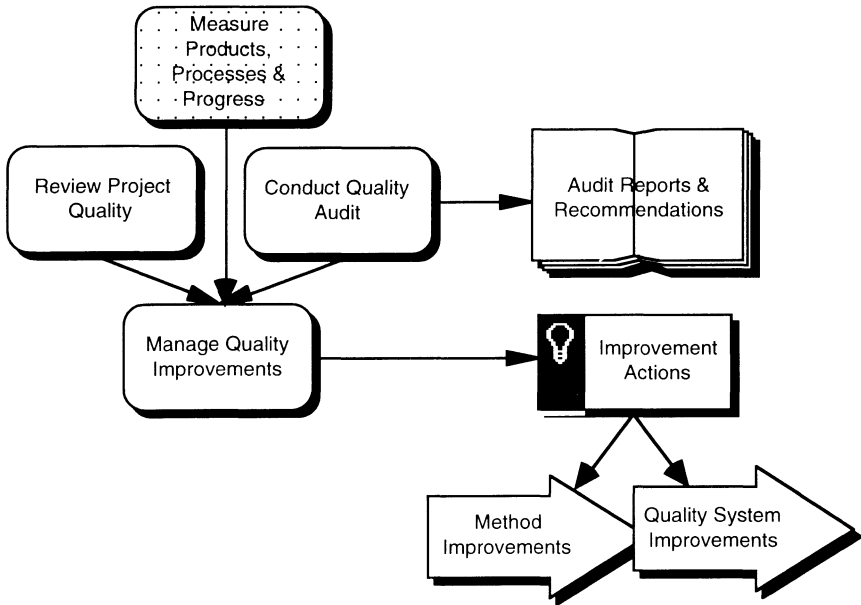


Figure 6 - Task structure: Improve Quality System

The key features of quality improvement are:

- a set of procedures for quality reviews and audits, both formal and informal
- the introduction of a quality improvement culture, empowering staff to deploy appropriate quality improvement tools.

IE\Q TECHNIQUES

Two specific techniques have been developed in conjunction with IE\Q. These are enterprise modelling and quality metrication.

Enterprise Modelling

Enterprise modelling is a way of clarifying the responsibilities of the agents within the scope of the quality system. It has grown out of a previous Information Engineering technique, known as RAEW analysis, which used a matrix to depict and recluster the Responsibilities, Authorities, Expertise and Work of each agent.



The primary constructs in the modelling language are AGENT, ACTIVITY, RESOURCE, STATE and INTENTION. Secondary constructs include EVENT, RESPONSIBILITY, OBLIGATION and CONTRACT. Expertise is modelled as a special kind of RESOURCE.

Quality Metrication

Within IE\Q, the principle is that quality metrication must be based on the quality drivers of the particular situation. IE\Q incorporates a framework for identifying and using appropriate context-sensitive metrics, based on the IEEE draft standard [IEEE P1061/D21].

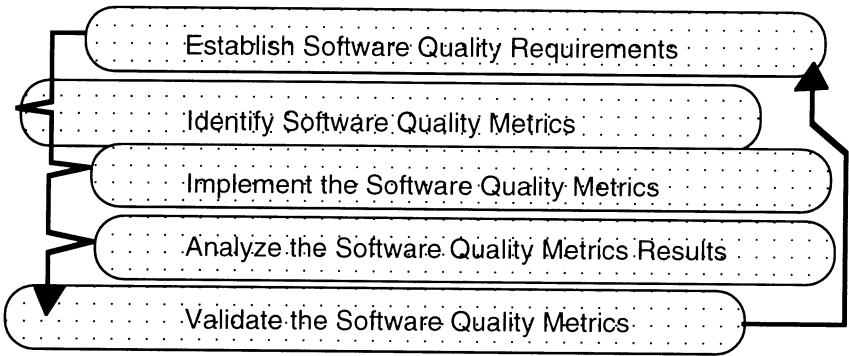


Figure 7 - Software metrics methodology (IEEE P1061/D21)

It is our intention to continue to track public-domain work in the metrics area, and to keep IE\Q up-to-date with future developments, as well as conducting further research and development ourselves.

QUALITY IMPROVEMENT IN IE\Q

In addition to the techniques described above, a portfolio of quality improvement techniques has been assembled from the quality literature.

Large numbers of quality improvement techniques exist in the literature. We have not created any new techniques in this area, but we have selected a useful set of techniques and classified them in terms of the role they play in quality improvement:

1. Techniques for finding problems / opportunities
2. Techniques for understanding problems / opportunities



44 Software Quality Management

3. Techniques for quantifying problems / opportunities
4. Techniques for finding solutions
5. Techniques for evaluating solutions
6. Techniques for implementing solutions
7. Techniques for organizing quality improvement

These techniques are applied incrementally, to enable the quality system to evolve to fit changing circumstances, and to take advantage of new opportunities and ideas.

SUMMARY

IE\Q is an application of engineering principles to the implementation of quality systems into the Information Management function. The intention is to minimize both the set-up cost and the operational costs of the quality system, and to ensure that these are more than covered by the benefits obtained. Two particular techniques have been introduced to fulfil these intentions: enterprise modelling, to clarify the responsibilities and obligations; and metrication, to match the costs to the benefits. In addition, techniques from the public domain have been included in IE\Q; these include techniques for incremental quality improvement, so that the quality system may enter a spiral of continual positive evolution. All these techniques are embedded into a task structure that enables the entire process to be effectively managed.

REFERENCES

Fenton, N.E., *Software Metrics: A rigorous approach* Chapman & Hall, London 1991

Frankson, M., & Vahee, G., 'Euromethod: towards a harmonisation of European methods' (undated paper)

Humphrey, W.S., *Managing the Software Process* Addison-Wesley, Reading MA: 1989

IEEE/P1061/D21, *Standard for a Software Quality Metrics Methodology* (Draft) April 1990

ISO 9000, *Quality Management and Quality Assurance Standards*, International Standards Organization, 1987



ISO TC184 SC5 WG1, *Framework for Enterprise Modelling*
International Standards Organization, May 1993

Texas Instruments, *A Guide to Information Engineering using
the IEF*, Texas Instruments Inc, Plano, TX, 1988

TickIT, *Guide to Software Quality Management Construction and
Certification using EN 29001* TickIT Project Office, 1992

Vahee, G., & Frankson, M., 'The Euromethod Structural Model'
(undated paper)

Veryard, R., 'What are methodologies good for?' *data processing*
27 (6) July/August 1985, pp 9-12