



A language for modelling quality processes

G. Born

*European Business Consulting Group, EDS-Scicon,
Pembroke House, Pembroke Broadway, Camberley, Surrey
GU15 3XD, UK*

ABSTRACT

Software quality management is shifting its emphasis from quality systems to quality processes. Ironically, this has resulted from the success of system-based initiatives such as Tick-IT, which emphasise the documentation and control of quality systems in software engineering. However, such approaches do not address the need to *improve* software quality, once the quality system is established.

To provide a basis for quality improvement, there is much work at present on understanding the processes of software development. Unfortunately, most of this has been a compilation of existing work, following a wide range of approaches, and without a standardised methodology for analysing and describing the processes which effect quality.

This paper describes a modelling language developed to avoid these problems. Based on a combination of dataflow and object-oriented techniques, QPL is a language built around the basics of quality management:

- Activities and tasks
- Authority and approval for actions
- Records and data
- Roles and responsibilities
- Conditions and constraints

Building on these simple concepts, a framework is built for the activities of quality management, including:

- Analysis of management processes
- Documentation of the quality system and authoring of standards and procedures

716 Software Quality Management

- Structuring configuration management and other record-keeping
- Reviewing activities and documents
- Checking for compliance with external standards, such as ISO 9000.

This framework also supports the automation of quality management, such as record-keeping, hypertext referencing of Standards, automated quality documentation and computer-assisted reviews.

INTRODUCTION

From Systems Thinking to Process Thinking

Software engineering is engaged in a revolution as significant as that of structured programming in the 1970s. Wherever one looks, there is activity into the nature of software production and its human contributors. Computers have not fulfilled their promise in assisting organisations and cannot, until their processes are radically re-engineered. Increasingly the emphasis is shifting from traditional systems analysis to enterprise modelling. The former concentrates on the computer system in isolation from the wider system of which it is a part, while the latter models all processes, human as well as computer.

While there is still much to do in applying ISO 9001 and other systems approaches to quality, the emphasis is shifting to a process-oriented view of quality. Examples abound, but the current interest in the SEI Software Capability Maturity Model and in the SPICE initiative [1] indicate trends for the rest of the decade.

Equally, many are dissatisfied with current system approaches to assessing software quality, in favour of software *product* assessment. Clearly, it is impossible to assess the quality of any product without understanding how it will be used. As with enterprise modelling, this requires an understanding of human as well as computer processes.

The traditional *systems* view of a computer system focuses on the final software product and places human elements at the periphery. The trends above are part of a shift toward a process viewpoint, where we model the activities and interactions of all agents in the systems, whether they are computer-based or human. Similarly, there is a trend toward modelling the *construction* of a large computer systems as a set of interrelated processes, where the human agents and their activities are the focus.

Process Modelling and Quality Management Need a Common Framework

With this trend, there is an obvious need to represent processes so they can be modelled. QPL - the Quality Process Language - provides such a framework by representing processes from the viewpoint of quality management. As this

paper will demonstrate, it is possible to combine process modelling with a representation of the activities which constitute quality management, including documentation of procedures, reviews, compliance and quality improvement. QPL provides a framework for this.

Contents of this Paper

This paper provides an overview of the QPL modelling notation, with a description of the elements of the language. It then describes ways in which QPL can be used for quality management activities. Finally, it provides examples of QPL applications, including software life cycle modelling, standards modelling and process enactment.

THE QUALITY PROCESS LANGUAGE: QPL

The General Approach

QPL concentrates on a process modelling view of quality management, although the technique is applicable to modelling software engineering and general business processes as well. The approach has been kept fairly simple, to make it accessible to a wide audience inside and outside software engineering.

The approach and notation are based on information flow, with some major differences:

- QPL puts much emphasis on definition of a process, which is called the *authority* for the process. An authority can be either written - such as a standard, procedure or directive - or qualified, such as an expert or consultant with appropriate qualifications.
- All processes must have ownership, represented in QPL by the *role*, who is responsible for all aspects of the process and the quality of its results.
- QPL defines three types of information, depending upon persistence: channelled information, which is transitory and only used for communication between processes; information stores, which retain information, but only the latest version; and archives, which retain one or more previous versions of the information. The foundations in quality systems are obvious, and these definitions allow the modelling of many configuration processes without getting into excessive detail.
- Object-orientation is used to allow processes to "overlap" through inheritance hierarchies. It can be used to construct hierarchies of standards and procedures, where the higher levels (such as international standards) are interpreted or activated by the lower levels (as in the case of company procedures). This is essential if one is to model standards, and the processes they regulate, in the real world.

718 Software Quality Management

Language Elements

There are five elements in QPL:

- Process
- Information
- Role
- Authorities
- Control

A **process** consists of a sequence of steps which transforms information from an initial state (input) to a final state (output). A key characteristic of a process is that it can be broken down into less complicated processes; QPL uses this to make processes manageable, by describing only a limited number of steps at any one stage.

A process cannot stand alone: it must have some input and some output. This information will provide the necessary links between processes and also provide the flow (or thread) of control which determines the order in which processes are carried out.

QPL does not deal directly with material objects (such as automobiles, components or software), but with information about such objects. It emphasises the description of these objects and the values that such a description may have. In the QPL notation, packets of information are called **information items** and are represented by the links which connect processes. QPL also distinguishes between information which is used by a process but not transformed, and information which is created or transformed.

Responsibility for a process is embodied in the **role**. It may be a person, in the case of most administrative processes or may be a machine - such as a computer for processing data or a robot arm for welding automobile parts.

The concept of *role* allows us to specify who or what is responsible for a process. **Authorities** provide an additional concept which makes clear how the process was carried out, and the basis for making decisions.

Authorities take many different forms. In a quality system, we find them most often in documented procedures and instructions, as well as in job descriptions and codes of practice. But an authority can also be "qualified", as in the case of a person using professional judgement, so long as the person or his/her qualifications are made explicit and documented. When we visit a doctor, we do not ask to see her written instructions for doing a diagnosis, accepting that her authority comes from a combination of qualifications and experience.

Control is a special type of information which is used to specify the conditions under which a process is activated. In QPL, control can be used for one of the following:

- The condition(s) under which a process is activated.
- The time when a process is activated
- Synchronising processes, using either of the above
- The constraints under which a process operates

Figure 1 illustrates the notation for these elements. Note that the position of an entry or exit on a process is significant: input always enters from the left, while output always exits from the right or bottom right. The control always enters from the top left, while the authority enters from the top right. The role is represented in the bottom of the process box; if it is variable, it enters from the bottom left.

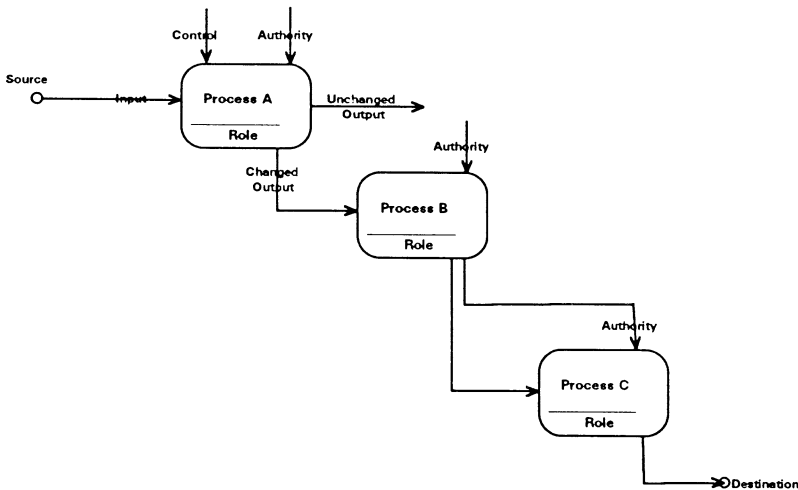


FIGURE 1 - THE QPL NOTATION

The figure shows a characteristic use of QPL: Process B is part of the quality management system and is used to create a procedure, which in turn is the authority for Process C.

USING QPL

Documenting Processes

The most simple use of QPL is to document processes. This is the equivalent of flowcharting or other diagrammatic techniques, but the language also provides additional features. Unlike a flowchart, it models information flow as well, thus enabling us to look at the transmittal and storage of forms, records and other documents. It also provides a mechanism for modelling the responsibility for all operations, and indicates the authority for all actions.



720 Software Quality Management

Because QPL is highly structured, it provides a representation which is the equivalent of complete procedures or work instructions. Thus, it can be used to represent the information in these documents in diagrammatic form. This enables us to review and criticise the processes more effectively than is possible for text. After this critical phase, the information can be transformed into the usual written documents, supplemented by diagrams if desired.

The object-oriented features of QPL enable groups of standards and procedures to be modelled as well. This is mainly done through the mechanism of the authority, which can inherit characteristics of other authorities. In a typical ISO 9001 organisation, the top authority would be ISO 9001 itself. Properties of that standard would be inherited by the Quality Manual and other high level standards for the organisation. Additional authorities would be provided by company-specific documents, such as policies and job descriptions. At lower levels, divisional procedures and codes of practice would inherit the properties of these intermediate documents.

Reviewing

Since QPL is designed to facilitate quality management, it can easily be applied to the activities of this discipline. Reviewing is a good example. Again, we take a strong process modelling position, and reviews are carried out by checking that the model conforms to a set of about 40 rules for well-formed processes. A violation of a rule corresponds to an observation in a conventional quality review and indicates an area where the process needs to be improved.

An excerpt from the list of the reviewing rules is shown in **Table 1**.

Type of Error	Description
Unconnected process	A process is not connected to any sources, destinations or other processes.
Hostile Process	A process has no inputs or control
Infertile Process	A process produces no outputs
Broken path	There is no possible path of information between a related source and a destination
Information validation omission	There are no checks to catch incorrect or incomplete information items
Process validation omission	There is no mechanism for catching or correcting an incorrectly applied process

TABLE 1 - EXAMPLES OF ERROR TYPES IN QPL

Compliance Checking

With the current emphasis on Tick-IT and third-party certification, quality management needs to pay attention to compliance with external standards, particularly ISO 9001. QPL supports this with an extension of its reviewing method.

The basic technique is to "wire-in" a model of the standard into the process modelled with QPL. In the case of a software producer checking compliance within the Tick-IT scheme, a model of ISO 9001 will be applied, and the process model generally will cover the standards, procedures, and codes of practice for the organisation. The result is a combined process model of ISO 9001 and the organisation's own processes.

Then the standard reviewing checks, described above, are applied. While other problems may be revealed by these checks (as in the case of a conventional compliance audit), these checks generally will reveal the areas where the organisation's procedures violate ISO 9001. Since the violations are related to an existing process model of the organisation, it is easy for it to identify the areas that need correction and then to modify the processes.

Process Improvement: The QPL "Life Cycle"

Finally, QPL also provides a route toward process improvement, with its own modelling life cycle. **Figure 2** illustrates this approach, which consists of a series of models. The **Template Model** is a model of a process to comply with, such as ISO 9001. We view this as a generalised template for any organisation; it is not intended as a specific process which could be tailored or extended to an organisation's specific needs. The **Foundation Model** is the model of the existing practices and procedures followed by the organisation, while the **Target Model** is the model of what the organisation would like to become at an identifiable horizon. While the Foundation Model is not compliant with the Template Model, the Target Model always will be.

These three models provide the beginning, direction and end to the change that the organisation wishes to accomplish. However, to provide support and markers along the way, two further types of model are provided:

- | | |
|-------------------------|---|
| Baseline Model | a stage in going from the Foundation Model to the Target Model, where achievement of this stage is capable of measurement |
| Transition Model | a process model for going from one baseline to another. |

722 Software Quality Management

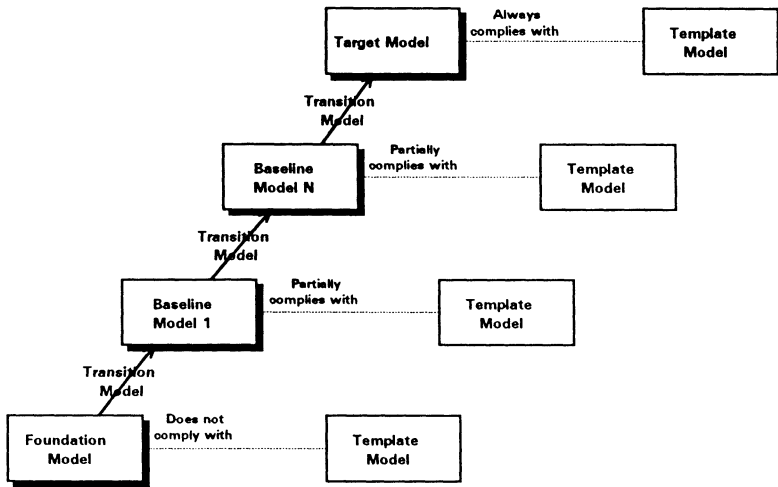


FIGURE 2 - LIFE CYCLE FOR PROCESS IMPROVEMENT

In terms of the SEI Capability Maturity Model (CMM), the CMM is the Template Model. The Foundation Model represents the current state of the organisation, while the Target Model represents the organisation it intends to become. The Baseline Models represent levels of Process Maturity, as defined by the CMM but applied specifically to that organisation. The Transition Model is the plan adopted by the organisation for moving between the levels of the CMM.

Currently, Template Models are available for ISO 9001 and the Malcolm Baldrige Award, which is widely used in North America. Further templates are planned for the future.

APPLICATIONS OF QPL

Quality Management

Most of the quality management applications of QPL were described in the last section. To summarise, they consist of:

- Design of documents in the quality management system, prior to their writing
- Documentation of organisational procedures, work instructions and codes of practice.
- Structuring of organisational structures and processes

- Quality reviews
- Compliance audits
- Process improvement

Business Process Modelling and Improvement

While this paper puts its emphasis on the use of QPL for quality management, it is also appropriate for modelling general business processes. In that way, it shares much in common with other process modelling approaches. It differs by enabling the quality aspects to be modelled explicitly; this challenges the usual and rather naive assumption that once a process is properly described, it will be carried out correctly. By using the reviewing checks of QPL, it is possible to predict where the vulnerable points in a business system will occur. A set of checks are provided specifically for FMEA (Failure Mode and Effects Analysis). It also provides guidance on improving the process so that weaknesses and sources of error are removed.

When applied to business process re-engineering, QPL provides additional reviewing rules which can be applied to the dynamics of process change. For example, one of the rules states that the authority must be *balanced* by the process it affects. This supports process change by concentrating on the practicalities of process implementation: when a procedure is rewritten, it obliges the process owner to ensure that staff are properly trained and provided with the resources needed to carry out the revised process.

Software Capability and Standards Modelling

At present there is much activity in the modelling of life cycles and in process improvement. For example, the plan for the SPICE Project [1] proposes to review current approaches to software process assessment and provide a method as a recognised international standard.

For this sort of work, a QPL representation of all relevant "standards" would be developed as Template Models. These *authorities* would then be connected into a model of software development. There are a number of possible models which could be used here, including the IEEE and ISO Life Cycle models (a Committee Draft at the time of writing).

The use of the standardised checks described above would then reveal where there are logical errors and inconsistencies. However, unlike the design of processes, this approach would be used to refine the various multiple authorities into a single process model representing the "ideal" mature process. After a stable process description was achieved, the process model for the authority would then be separated from the Life Cycle model and used to develop checklists and other practical aids for assessing capability and maturity.

724 Software Quality Management

QPL can also be used to check whether standards contradict one another and can provide a framework for positioning the scope of working groups tasked with the creation or revision of international and national standards.

Process Enactment

For several years, EDS-Scicon has provided computer systems which automate processes within quality management. In particular, the Q-Doc system provides integration of three main capabilities [2]:

- An Expert System, which contains and processes specialised knowledge of quality management and how it is to be applied to a problem
- Hypertext support, which enables users to access information in standards and procedures quickly
- Document Generation, which writes large documents using the knowledge in the expert system

Figure 3 shows the architecture of the Q-Doc system

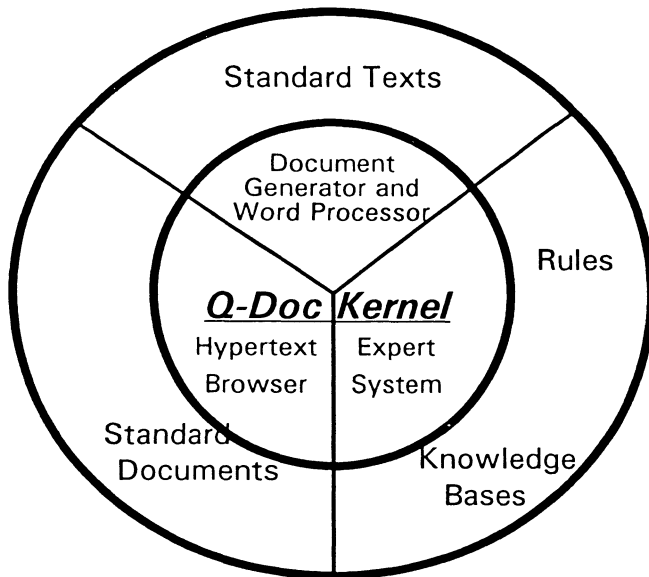


FIGURE 3 - Q-Doc ARCHITECTURE

Within EDS-Scicon, the Q-Doc system has been used to produce quality plans for projects and to assist project managers in understanding company requirements. This required a model of all management functions associated

with a project, from project and financial control to life cycle planning and execution. Applying conventional systems analysis and design techniques was sufficient to produce this application, but we found it very difficult to model certain aspects of quality management, especially how to attach general standards and procedures to specific project processes.

The QPL notation provides a better way to model these processes, so that they can be enacted in a computer system. In particular, the use of authorities not only provides an elegant mechanism for specifying which standards and procedures apply to each activity within a process, but also determines the hypertext links. The authority, which is part of the QPL description of each management activity, tells the system which document in the quality management system should be displayed, when the user requires further guidance about that activity.

As workflow and other process enactment systems become more widely used in the future, QPL will provide a useful mechanism for recording and analysing the processes which they incorporate and animate.

ACKNOWLEDGEMENTS

The Q-Doc system was developed by EDS-Scicon. The author is grateful to Tom Abram and Maurice Resnick, who provided support for the production of this system. Jeff Hiscock and Andy Tomiak developed the Q-Doc system and its first applications.

This paper is the sole responsibility of the author, and the opinions expressed in it are his and not necessarily those of the EDS Corporation.

REFERENCES

1. Peter Simms, ed., *Study Report: The Need and Requirements for a Software Process Assessment Standard*, Issue 2.0, June 1992. Published by ISO/IEC JTC1/SC7/WG7/SG1.
2. Gary Born, "An Expert Support System for Software Quality Management", 3rd European Conference on Software Quality, Madrid, November 1992.