ENGINEERING THE BUSINESS THROUGH QMS DESIGN

Authors: Bernard Hawkes & Rhys Williams

Date: 24th October 2006

TABLE OF CONTENTS

INTRODUCTION

As organisations become more complex, so do their Quality Management Systems (QMSs). In large organisations, it is no longer sufficient to produce a quality manual and a library of procedures. Rather, the QMS needs to be designed using the same principles that are used to design any other system.

In addition, the QMS should be clearly focused on the business and its needs, rather than just aiming to implement a pre-defined external model. This implies that the QMS should be based around a model of the organisation's key business processes.

When properly defined, the business processes and the roles defined within them become drivers of the organisational structure, rather than being dictated by it.

What is involved in developing or improving a QMS should not be underestimated - it is not a matter of just writing quality procedures, it is about engineering the whole business.

This white paper takes a look at the importance of designing a QMS rather than just implementing a set of business processes, aligning the QMS to the business model, the architecture and attributes of the QMS, and effective process documentation.

The aim is to raise the reader's awareness of what is required when designing and implementing the QMS, rather than stating how it should be done.

THE IMPORTANCE OF SYSTEM DESIGN

For a small organisation, the QMS may simply be a set of procedures brought together by a quality manual. For larger organisations, the QMS may need to deal with such things as:

- Different geographical locations.
- Diverse types and styles of working.
- Complex organisational structures.
- Multiple methods of presentation and distribution of the QMS.
- Language differences.
- Differences in the legislative environment.
- Large numbers of elements making up the QMS.

In such cases, the word 'system' takes on its full meaning. Like any complex system, a QMS needs to be designed *before* it is implemented. This does not mean what the documents will look like (although this is, of course, important) but determining what all the elements of the system will be, the interfaces between them, how data will be managed and what the user interface will look like - in fact all the aspects which would be considered in any normal system design process.

Also like any other system, a QMS must be designed to satisfy a defined need. This implies a QMS design (or re-design) project should not be embarked upon until the requirements for the system have been defined and agreed. Although this seems obvious, it is often overlooked. At one end of the scale, this can lead to unrealistic expectations of what the QMS will achieve. At the other end, it can lead to important aspects of the organisation being placed 'out of scope' of the QMS in an ill-considered manner.

Change is the only thing that is certain in modern organisations. QMSs should, therefore, be designed in a way that anticipates change and minimises the need to re-work the system each time a change occurs.

DEFINITION OF 'PROCESS'

Before continuing, it is worth stopping to consider what is actually meant by the term process. Many heated discussions often arise about whether something is a process, a procedure or something else.

The term process was first popularised, in this context, by the proponents of Business Process Re-engineering (BPR). There are many definitions of 'process' in the BPR literature. The key aspects of a process are that it:

- Is a set of activities.
- Takes inputs and converts them to outputs.
- Adds or creates value for the customer (recipient of the output).
- Focuses on how the task is performed, as opposed to what is produced.

Other important characteristics of a process are that it should:

- Have an owner or champion.
- Have a beginning and an end.
- Be repeatable (which normally means it is documented).
- Be supported by appropriate tools and training.

BRANDING

Shakespeare wrote:

"*What's in a name? That which we call a rose by any other name would smell as sweet."*

As far as QMSs are concerned, the name they are given means a great deal. The word 'quality' has a large amount of baggage associated with it - some good, much not so good. Many people consider the word to put a boundary around the system, making comments such as "That shouldn't be in the QMS" or, even worse, "Well of course, the work I do has nothing to do with quality". (*This is a real quote from someone in the finance function of a large organisation*.)

Finding a name for your QMS is strongly recommended, preferably one that does not generate its own baggage. Aside from eliminating the de-scoping problem mentioned above, a name allows a level of branding to be achieved. If the name is combined with a logo or some other graphic device, this makes the branding even stronger. Branding in this way also helps top management to see the QMS as a valuable product, rather than a necessary overhead.

THE UNDERLYING MODEL

The design of the QMS should be based on an underlying model of the business. Although all of the following have slightly different purposes and approaches, they all provide some form of model or framework:

- ISO 9000 series
- TickIT
- ITIL
- Six Sigma
- EFQM Business Excellence Model
- SEI's Software Capability Maturity Model (CMMSM)

Numerous other approaches could be added to the list and all have their own strengths and weaknesses. All have their fair share of success and horror stories. So how do you choose which to use?

It may be heresy, but one suggestion is that none of these models are used as the sole basis of the QMS design. The reasons for this are:

- They are all external to the organisation, so need to be sold internally.
- They all need tailoring to the needs of the organisation and its business.
- Too many of them are focused on assessment of the system, rather than its original design.
- In a large organisation with a diverse range of customers, it is likely that different customers will be familiar with, or even require, different models.

The best and most appropriate concepts from all the models can be used as a basis for designing a business model and system architecture specific to any given organisation. The resulting design then 'looks and feels' like the organisation. It is much more readily accepted than something that needs translating from some external language.

If it is then required to compare or assess the system against one or more external models, this can be achieved by providing mapping matrices.

BUSINESS MODEL

The business model shows all of the main organisational processes in a single, normally graphic, representation. A recommendation is that it is produced in a conceptual manner, rather than trying to make it an absolutely precise model of the organisation. The latter is useful to the designer and can be produced as part of the system design, but it is usually far too complex to be useful to the average user.

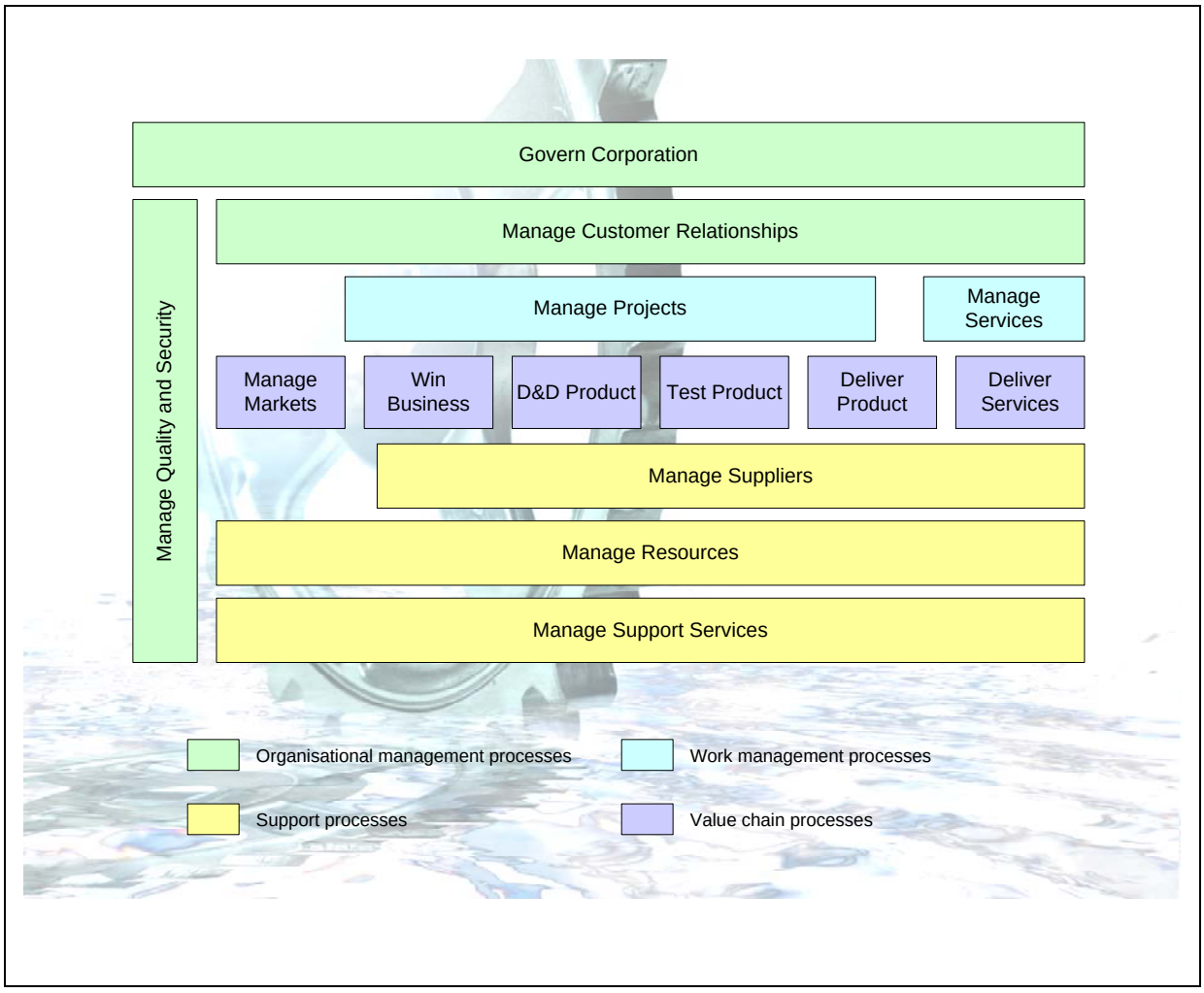

Figure 1: Conceptual Business Model

Figure 1 (above) shows an example of how the business model can be presented. (Note that this is intended to demonstrate the concept, rather than to provide a complete model.)

SYSTEM ARCHITECTURE

The system architecture should not be confused with the business model. It is about the elements of the QMS, their scope, and how they relate to each other. The following are elements of a possible architecture, shown graphically in figure 2.

Policies

The quality manual may contain policy statements, such as the quality policy and the environmental policy. However, as the quality manual is usually produced professionally for customer consumption, it is wise to avoid including anything that may change frequently and, therefore, require constant updating. It is also often the case that some policy statements need to be visible to the customer whilst others are for internal consumption only. Policy statements, therefore, benefit from being individually controlled and collected together in a policy manual.

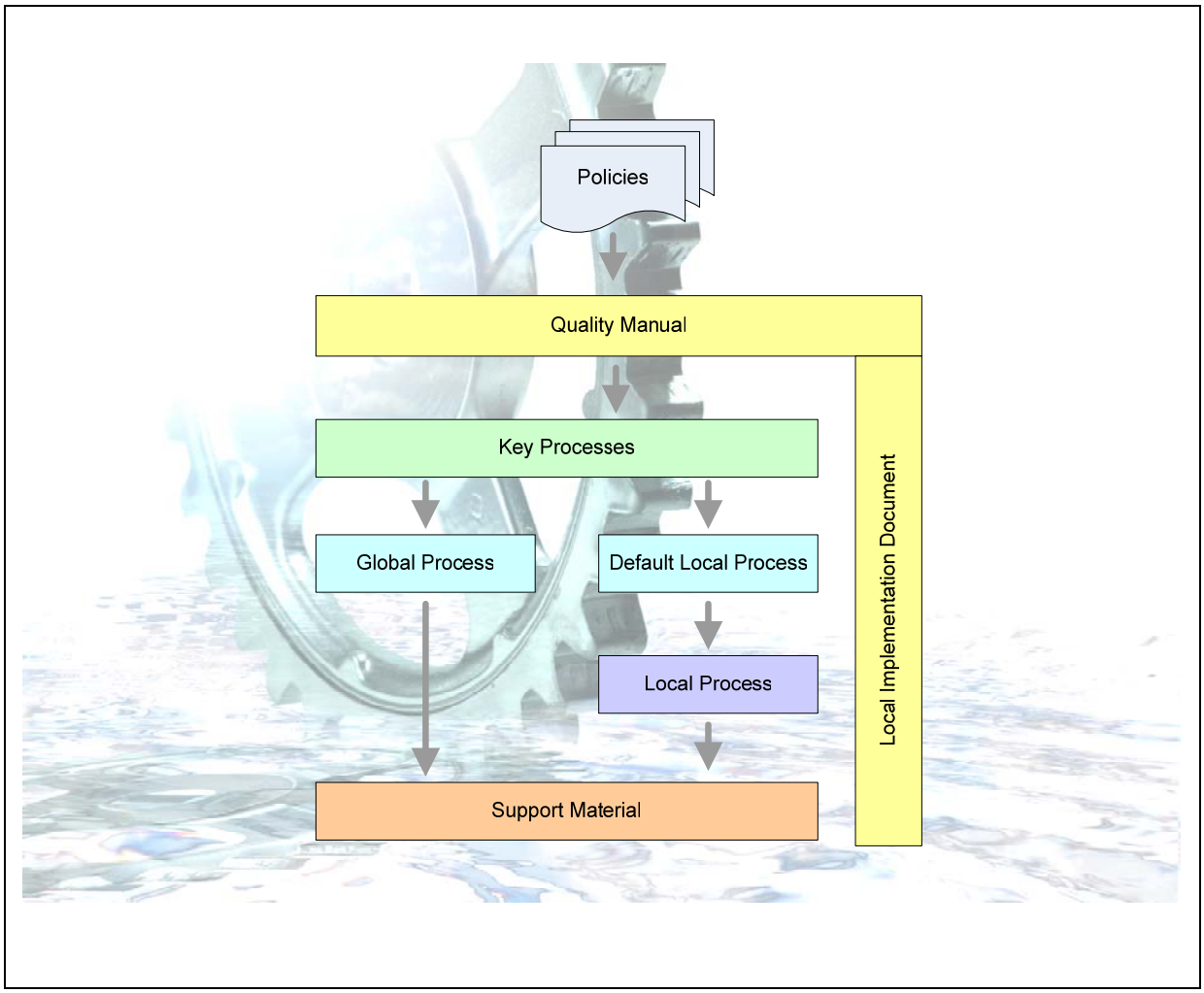

Figure 2: System Architecture

The (Quality) Manual

Again, removing the word quality from the title takes with it a great deal of baggage. The manual contains:

- A statement of the organisation's policy on, and commitment to, quality.
- The business model.
- The system architecture.
- Definition each of the key processes in the system.
- The main roles and responsibilities within the organisation.

Key Processes

As mentioned above, the business model defines the key business processes for the organisation. Each of these is then expanded in top-level process descriptions.

The aim of these key process descriptions is to provide a consistent framework for processes across the organisation. They should be written so that they are independent of local variations in the organisational structure and of any tools that are used to support or implement them.

Key process descriptions define *what* (as a minimum) the organisation expects. It is recommended that each key process has a set of business rules for the process - these should be minimal. (Avoid putting in an extra rule every time something goes wrong!) For example, many organisations will have a 'Manage Projects' business process. This will have rules such as:

- Every project must have a designated manager.
- Every project must have a documented plan.

These business rules, together with the objectives of the key process defined in the quality manual, provide the requirements for the process. The detail of the process then goes on to expand upon these.

So, to continue with the example, the 'Manage Projects' key process will probably have within it sub-processes for 'Select and appoint project manager' and 'Plan project'. (Note the fundamental difference between 'Plan project', which is a process, and 'Produce project plan', which is an activity. Many project plans appear to be produced without the benefit of any defined planning process.)

As key processes are intended to be at a high level, many of the activities defined in them will be processes in their own right and will therefore need to be defined separately.

Process Descriptions

Process descriptions at this level should define *how* the job is to be done, rather than what is to be done.

These could be referred to as sub-processes. However, this becomes very cumbersome, particularly if a further level of detail (sub-sub-processes) is required. It is sufficient to call them all processes, allowing any process to be defined in terms of lower level processes until, at the lowest level, these are replaced by activities. Some care should be taken to avoid continually leaving detail to lower level documents and thus creating massive, and unusable, hierarchies. However, 'library' type processes (for example 'Review document') ideally fit into a lower level of the hierarchy since they may well be called into play from several other processes.

Support Material

This is everything else which is not specifically a process. Organisations may choose to divide support material up into further categories, particularly within their document numbering schemes. Included under this heading are such things as templates, forms, descriptions of techniques and checklists. One important class of material included at this level is the set of documents that define how to use a specific tool to execute one of the defined processes. (Note this is not the same thing as a User Guide for the tool.) Keeping the process-tool interface at this level allows the tool to be changed, or different tools to be used in different areas of the company, without a major impact on the core elements of the system. This does involve taking some extra time to write the related processes generically but this is well worthwhile.

Roles

Processes necessarily define who is responsible for each particular activity. To maintain organisational flexibility, it is useful to use role names, as opposed to job titles, in process descriptions. For example, 'Plan financial authority' is a role whereas 'Departmental manager' is a job title. Sometimes these will coincide - 'Project Manager' is usually a role name and a job title.

Roles can then be assigned to individuals in a one-to-one, one-to-many or many-to-one fashion. Such assignments can be part of the local implementation documentation or individuals' job descriptions. This allows organisational differences, and changes, to be supported with minimum impact on the processes themselves.

PROCESS DOCUMENTATION

There are numerous ways to document processes, some clearly defined and some less so. The starting point must be a graphical representation of the process. Role Activity Diagrams are the best way of doing this from a practitioner's point of view. However, they tend to be complex for the end user.

In an IT environment, simple flowcharts are easy to draw and are easily understood. They are excellent for defining process flow, but not very good at dealing with information flow, so this needs to be considered separately.

The body of the process should have a section corresponding to each 'box' on the flowchart. For the most common flowchart elements, the corresponding section answers the following questions:

- Start box: What triggers the process? When does the process start and what is needed to start it?
- Activity box: What are the inputs and outputs? What has to be done to carry out the activity? Who does it and when? What checks, measures and records are required?
- Decision box: Who makes the decision? What are the criteria on which the decision is based? What are the possible outcomes and what happens next (in the process) depending on these outcomes?
- End box: When does the process end and what is needed to end it?

Presentation

Although this is a subject in its own right, it deserves a brief mention.

There is always a trade-off between three different views.

- The process practitioner requires something that is good for defining and maintaining sets of processes; ideally, purpose-designed tools with graphical interfaces.
- The author (the subject matter expert who is going to provide the main input to the process) requires something familiar and easy to use, such as a word processor.
- The end user requires something that is easy to access, search and follow whilst carrying out routine tasks. An example of this is intranet-based presentation.

QMS documentation is sometimes maintained in several different forms to satisfy all these requirements. This is clearly a maintenance nightmare. The rule has to be that there is only one source for any given piece of information.

One very good solution to this problem is to maintain all the elements of the QMS in SGML (Standard Generalised Mark-up Language) or XML (eXtensible Mark-up Language). Whilst it is outside the scope of this paper to explain fully, the SGML approach allows a single source of information to be presented in various different ways automatically. This is an approach that is well worth investigating before embarking on a major new QMS documentation programme.

Language

Whatever the presentation mechanism, it is almost certain that a good deal of the material will be plain text.

Many quality documents seem to be written in a language of their own (QA-speak?). Terms such as 'documentary evidence' and 'timely corrective action' abound in a way that is totally inconsistent with normal English grammar. It may all be very well for international standards, but QMS documents are supposed to be written for normal people, not quality professionals. They should, therefore, be written in normal English.

The excellent guidance provided by the Plain English Campaign is recommended as it helps in producing clear, concise and easy-to-read language.

DEALING WITH LOCAL VARIATION

Even in large organisations it is desirable to use the same QMS throughout the organisation. This provides consistency in operation as well as supporting staff movement and organisational re-structuring.

In reality, there may be significant local differences already discussed above. Inevitably, there is a need for enough flexibility in the system to allow for local variation. In the scheme described above, key processes are mandatory throughout all parts of the organisation. However, other processes can be defined as either global or local. A local process can then be changed to suit the local factors. This may involve anything from a straightforward translation into another language through to a complete re-write. Where local processes are allowed, ideally there should be a 'default' process defined, which is used by any part of the organisation that does not need to produce its own local variation.

Something is required to define exactly what elements of the system apply to a given part of the organisation. Rather than having a multitude of different quality manuals to do this, a local implementation document is recommended. This defines how the key processes are implemented, by reference to the appropriate local processes. In addition, the local implementation document can add local business rules to the key processes (although it should not contradict those in the key process itself). The local implementation document also contains the mapping from roles to organisation.

SUMMARY

To summarise, when considering the need for a QMS, the requirements should be defined and agreed first - the system is then designed to meet them. This ensures that the QMS is designed to satisfy a defined need, a point which is often overlooked.

The importance of branding the QMS should also be considered. This allows the baggage associated with 'quality' to be shrugged off and also helps management to see the QMS as a valuable product rather than a necessary overhead.

The design of the QMS should be based on an underlying model of the business. The result being a system that 'looks and feels' like the organisation and which is more readily accepted by the users.

The QMS architecture should be carefully planned as it defines the elements of the QMS, their scope and how they relate to each other. There should be enough flexibility to deal with local variations - large organisations spread over wide geographical areas will almost certainly require this flexibility.

Finally, process documents should be as simple as possible, being produced for the end users rather than the quality team. The documents should have a common look and feel and be written in plain, simple English.

Hopefully, these points will provide food for thought and provide the reader with a good idea of the benefits that can be obtained from the introduction of a well-designed QMS.