

11 DOCUMENTING AND MANAGING REQUIREMENTS

Debra Paul

INTRODUCTION

This chapter is concerned with two of the key elements of requirements engineering – documenting the requirements that have been gathered and managing those requirements in such a way that they can be traced through the business change process, from source to delivery. Documenting requirements clearly is vital for the success of a project. Many projects have failed because of they have lacked well-formed definitions of the requirements.

THE IMPORTANCE OF DOCUMENTATION

There are many reasons for needing good documentation. First, it enables communication within the project team and provides a basis for ensuring that all of the related requirements are consistent with each other. Second, the documentation provides the business managers and staff, who are the sources and owners of the requirements, with a firm basis for validating that the documentation accurately records what they need the solution to provide. Third, any further work to develop and test the business solutions will use the documentation as input to these activities. The requirements documentation will define what the solution is to do and the acceptance criteria required to test that the required features have been delivered correctly. The requirements documentation is also used following the implementation of the solution. For example, in the maintenance of IT systems and during benefits realisation.

THE REQUIREMENTS DOCUMENT

Structure

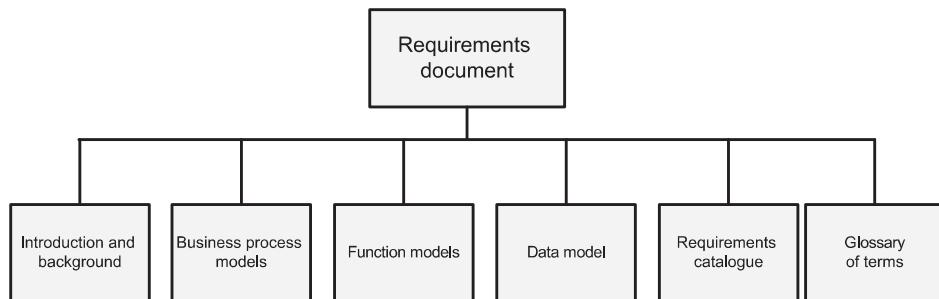
The requirements document has to provide the basis for the solutions to be delivered to the organisation so it needs to be well formed and clear. The business managers and staff have to review the documentation in order to ensure that the descriptions of the requirements reflect their needs and that the analysts have correctly interpreted and understood the information provided to them. One of the key tactics to support the review of the documentation is to ensure that it is organised and well structured. The requirements document needs to provide all of the information that the reviewers require in an easily understandable and digestible form. A well-structured document will help to improve the accessibility of the document and enable reviewers to identify

errors and omissions. The requirements document may be partitioned by requirement type, business area or, at a more detailed level, for specific groups of stakeholders. Whatever the basis used for partitioning, an organised requirements document will make reviewing much easier and help to ensure that the agreed requirements are accurate.

Content of the requirements document

The requirements document will usually contain the sections represented in Figure 11.1.

Figure 11.1 Contents of a requirements document



- **Introduction and background.** This section should set out a description of the business situation and drivers for the project. It serves to clarify the objectives and scope of the work and ensure that all stakeholders are aware of the business context for the requirements.
- **Business process models.** Typically, business requirements involve changes to the business processes and any new or enhanced software solution must support the business process changes. In this section the 'to be' process models should set out the vision for the new processes and may be accompanied by more detailed task models. The 'as is' processes that are to be revised may also be included here if required for additional clarity or explanation.
- **Function models.** Diagrams showing the functionality of the proposed software solution may be included here. Typical diagrams used here are context diagrams and use case diagrams. These diagrams provide an excellent overview of the solution and provide a means for structuring the requirements catalogue. Context diagrams and use case diagrams are described in Chapter 12.
- **Data model.** A data model is highly relevant where the requirements require a detailed level of data definition, for example, if it is intended to build a software solution or evaluate an off-the-shelf software package. However, it is also useful to develop a data model to understand the nature of the data and the business rules that apply to the relationships between data groups. If the requirements catalogue contains lists of data within individual requirements, these lists are extremely difficult to review and evaluate for correctness. Also, the structure of the data, and how the different groups of data need to relate to each other, cannot be defined easily within a textual description. A data

model provides a far superior view of the data than textual descriptions and enables the analyst to consider the exact business rules and requirements to be met by the solution. It is also a parsimonious view that enables the analyst to ask searching questions. Two data modelling techniques, entity relationship modelling and class modelling, are described in Chapter 12.

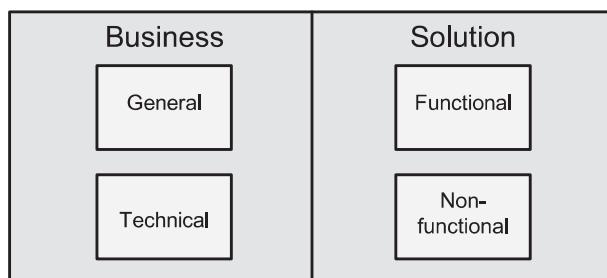
- **Requirements catalogue.** The information about each individual requirement should be documented in the requirements catalogue. The catalogue is the key component for the audit trail of the requirements as it is the central repository of information relating to the identification, cross referencing and source of the requirements. This document is described in more detail below.
- **Glossary of terms.** One of the key quality characteristics for the requirements document is to ensure that it provides a clear definition of the requirements so that it can be read, understood and agreed as easily as possible. However, within any organisation there will be terminology that is understood by the people working in a particular area and often this terminology is very precise in conveying information. As a result, while it is important that the requirements use this terminology, this can present a problem for the analysts and reviewers who lack familiarity with the terms. A glossary of terms overcomes this problem and provides a central source of terminology definitions. The glossary may be created just for a particular project or there may be an organisation-wide glossary that can be used as a basis. Either way, it is an important component of the requirements document and will be used throughout the business change and software development lifecycles.

This is not an exhaustive list and many organisations include other areas in their templates. For example, a list of assumptions or decisions made can often be found in a requirements document.

THE REQUIREMENTS CATALOGUE

When requirements are initially elicited, they are not organised and it is only once the requirements analysis activity takes place that they are structured and formed into an organised set. There are a number of ways of organising requirements but fundamentally, a hierarchical approach will provide the easiest structure for navigating and reviewing the requirements. Figure 11.2 shows the four types of requirements: general, technical, functional and non-functional. These categories provide a useful overview structure for a requirements catalogue.

Figure 11.2 Types of requirements

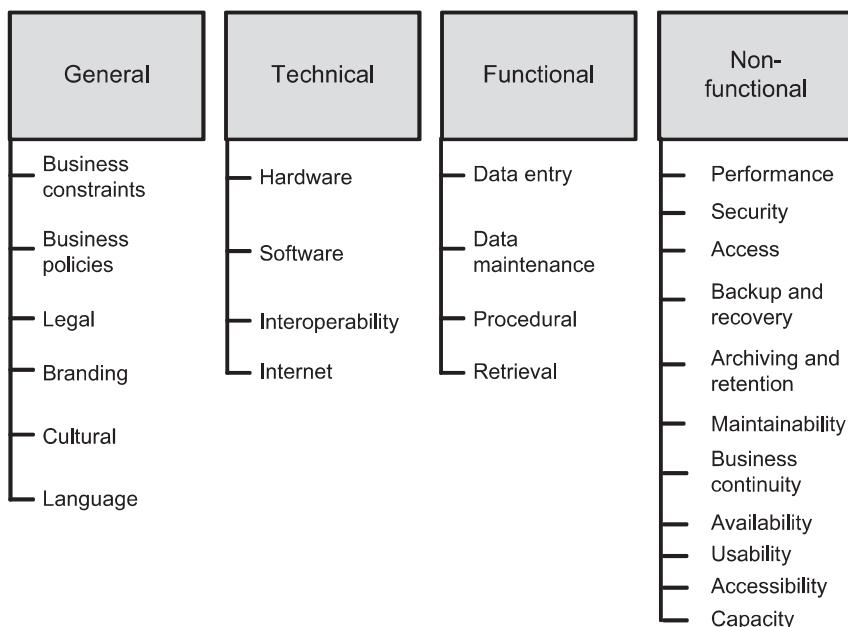


This hierarchy structures the requirements into four discrete areas, based upon the type of requirements. These are described below. At the next level, the hierarchy uses other subdivisions to categorise the requirements. For example, within the functional requirements, all of the requirements relating to reporting may be grouped together or each use case may be used as a mechanism to group requirements.

Types of requirements

Within each of these areas there are several specific categories of requirement that should be considered in further detail. We can use these categories as aides-memoire to ensure that areas of requirement are not missed. As discussed in Chapter 10, tacit knowledge is a major issue when eliciting requirements and the categories of requirement provide a useful basis for asking specific questions about areas that the business staff may not consider relevant or are taken for granted. Figure 11.3 shows the key categories of requirement for each type identified above.

Figure 11.3 Categories of requirements



General requirements

These are the requirements that define business policies, standards and needs. These requirements are often very broad in scope and can have an impact upon a number of different areas. Many general requirements apply to entire business change programmes, sometimes all of the change initiatives underway across the organisation. There are specific sub-categories of general requirement:

- **Business constraints** – these cover aspects such as budget, timescale and resources.
- **Business policies** – these cover aspects such as standards and business policy decisions (often known as business rules). These policies and standards ensure consistency of operation across the organisation and are often linked to the vision and values of the organisation. Business policies may define specific areas such as the environmental policy or the strategy to be adopted to ensure business continuity.
- **Legal** – these are the requirements that state relevant legal and regulatory constraints. These requirements may relate to organisations in many business sectors, for example the Data Protection Act applies to any organisation that holds personal data, or may be specific to a business sector or industry, for example, the Sarbanes–Oxley Act of 2002 and Basel III Accord which apply in the financial sector.
- **Branding** – these requirements are concerned with the image and style to be promoted for the organisation. Typically, there will be branding documentation, for example a style guide, which will set out factors such as logos, key words, language and colour requirements. This documentation will ensure a consistent brand and image are established across all forms of communication deployed by the organisation. The style guide will set out the 'look and feel' of any systems and documents used within the course of the organisation's work.
- **Cultural** – these requirements relate to the type of culture required within the organisation. This may mean that the requirements set out the vision for the organisation, the approach taken to dealing with customers and the management style.
- **Language** – many organisations have specific language requirements as they are operating across international boundaries and languages. These requirements set out the languages to be used in the organisation and the ways of communicating with customers and other organisations.

Technical requirements

These are the requirements that state the technical policies and constraints to be adopted across the organisation and apply to a range of change projects. These requirements may refer to the artefacts that describe the technical infrastructure for the organisation. The specific sub-categories of technical requirements are:

- **Hardware** requirements covering aspects such as the make and model of hardware equipment to be used in the organisation. The requirements may cover the IT hardware but can also include equipment relating to the work of the organisation such as production or general office equipment.
- **Software** requirements covering areas such as operating systems, software applications, networking and communications software. There may also be standards for use of online software such as cloud computing services.
- **Interoperability** requirements that cover the standards for communicating between systems where they are required to exchange data. These requirements may also need to be specified where systems need to communicate with other

technical equipment such as printers. The interfaces may be with systems and equipment operated within the organisation or by other, external organisations.

- **Internet** requirements. These requirements relate to the technical policies governing the organisation's use of the internet and web-enabled services.

Functional requirements

The functional requirements set out the features that any solution should provide. The features set out in these requirements may be provided by an IT system solution but this is not necessarily the case. Some of the requirements may prove to be too costly or time-consuming if provided by IT, in which case alternative means of delivering them will need to be considered – or they could be dropped altogether. Some requirements may be better delivered by non-IT means, for example, a personal communication may be more effective.

The functional requirements form a key element of the requirements catalogue because they define the specific features to be delivered by a solution. Also, whereas the general and technical requirements tend to have a certain longevity, the functional requirements can be subject to frequent change, particularly as they are elaborated to a more detailed level. The level of detail used to define the functional requirements, and when this detail is completed, will depend partly upon the delivery mechanism adopted. For example, if a linear software development approach is required, the requirements will need to be defined in sufficient detail for the design and development of the IT system to be undertaken. Alternatively, if an iterative development approach is to be adopted, the detailed requirements may evolve during the development activity and the documentation updated at a later stage. However, sometimes, the level of the requirements documentation to be produced is not this clear cut. For example, in situations where the development is outsourced, an iterative approach may be taken but the requirements will need to be defined to a sufficient level to help manage any communication difficulties. Also, it may be the case that while some of the requirements may evolve during the iterative development, there may be a part of the system that requires the application of complex business rules; in this case, it would be very sensible to define these clearly up front.

If the requirements are to be managed closely, certain characteristics – such as the identifier and source – need to be documented so that each requirement can be tracked through the development and delivery of the business solution. When eliciting and investigating functional requirements, it can be useful to consider several categories:

- **Data entry** requirements are concerned with gathering and recording the data that is required in the solution.
- **Data maintenance** requirements handle changes to the data used within the solution.
- **Procedure** requirements. One of the key sources of error in a business solution is where the detailed business rules to be applied during the working procedures are not understood well. These requirements need to be defined clearly so that they can be adopted accurately within the solution. Techniques such as decision trees or decision tables can be very useful in documenting these requirements.

- **Retrieval** requirements. These requirements are concerned with requests about the data and include the provision of specified reports and responses to enquiries. Management and operational information requirements can be numerous and wide-ranging. In this case, it can be useful to consider these requirements as a separate area of functionality when structuring the requirements catalogue.

Non-functional requirements

The non-functional requirements are concerned with how well the solution will operate and answers questions such as: 'how quickly will it respond?' and 'how easy will it be to use?' There are several areas of non-functional requirement to consider:

- **Speed of performance.** These requirements concern the speed with which a transaction should be processed. For example, if a customer wishes to order some goods or place a booking, this requirement would define the speed of the processing to handle this.
- **Level of security.** The majority of organisations handle information and data of varying levels of confidentiality. These requirements identify the security levels required for the organisation's data. The security levels are likely to differ depending upon the nature of the data. Some will be highly confidential and require extremely rigorous security while other data may still be confidential but subject to less security. The storage requirements for the data may also be defined here.
- **Access permissions and constraints.** These requirements are usually related to the security requirements as they define the stakeholder groups and their levels of access to the data. The access permissions will state which stakeholders are able to carry out which transactions. For example, which stakeholders are allowed to delete the details held for a customer or pay an invoice from a supplier. The requirements relating to the provision of functions to implement the access permissions will be documented within the functional requirements.
- **Backup and recovery.** The requirements for secure storage and access of information are also linked to these requirements. While it is important to guard against confidentiality breaches, retaining data is also vital, particularly with the rise of remote storage and cloud computing. The backup and recovery requirements define the policy for protecting against the loss of data and information.
- **Archiving and retention.** The retention of data and information retained within an organisation may be subject to internal policies or external legal regulations. These requirements define aspects such as the length of time of the retention, the nature of the archiving approach and the approaches to be taken to the disposal of information and data.
- **Maintainability.** These requirements are concerned with the approaches to be taken to maintaining the solution. These will include aspects such as servicing, problem investigation and correction.
- **Business continuity.** The ability of the organisation to continue to function in the face of various threats, natural and man-made, is very important and gives

rise to various disaster recovery and business continuity requirements. For example, there may be requirements which state how quickly an organisation should be operational, possibly in a limited form, following a business continuity incident. Business continuity requirements may include those to prevent a disaster affecting the organisation to any great extent – for example, the need for duplicate installations – and also contingency requirements should a disaster occur. There are likely to be several non-functional requirements related to the business continuity requirements, for example, in the area of backup and recovery.

- **Availability.** The timeframe during which a solution must be available to stakeholders. For many web-enabled systems, this is likely to be 24/7 (24 hours a day, 7 days a week) availability. However, other solutions may not require this level of availability or there may be some aspects that can accept a lesser level of availability. For example, a telephone enquiry service may need to be available from 8.30 am to 6.00 pm each day but should be supplemented by a recorded message service outside these hours.
- **Usability.** This area concerns the ease with which a stakeholder can learn, apply and use new processes and systems. It is a critical aspect of many IT solutions in particular because of web-enabled information and purchasing services offered by many organisations. Whereas internal business stakeholders can be trained to use new processes and systems, this will not be possible for external stakeholders such as customers so ease of learning and use is very important. There are many aspects that can be defined for usability requirements including speed of learning to use the system and ease of navigation (for example, the number of clicks to obtain information).
- **Accessibility.** The accessibility requirements are related to the usability requirements but they focus specifically on enabling access for all, regardless of infirmity or disability. We need to consider four aspects:
 - cognitive disability;
 - motor disability;
 - hearing disability;
 - visual disability.

These requirements state the need for features that enable accessibility for anyone with a disability. The means of meeting accessibility requirements can be numerous and varied. For example, providing software facilities such as screen reading tools and using images rather than text, or physical facilities such as access ramps and lifts, or considering the height and positioning of buttons (such as those in lifts) and counters.

- **Capacity.** These requirements cover areas such as the volumes of data or images to be stored, the volumes of transactions to be processed and the number of stakeholders to be supported.

The non-functional requirements are areas that are often left until later or even dismissed at an overview level without clear thought and analysis. This can be a critical error. There are numerous apocryphal tales of organisational disasters – or near

disasters – resulting from such relaxed thinking. Public sector organisations have been criticised heavily for losing confidential data, or making it accessible when this should not be the case. However, private sector organisations are not necessarily any better at this. Some commercial organisations have promoted new services and then not had the capacity of staff and systems to handle the level of interest generated. Some websites providing online information and services are shockingly unusable. How often have you entered data only to be told after submitting the data that you have made a simple error and now have to re-enter the entire set? How often have you left an organisation's website in frustration at the lack of assistance and the time required to complete a simple enquiry? And accessibility for all is growing in importance with the increasing recognition of the need to ensure anyone with a disability has the access they need.

Business analysts have a range of techniques at their disposal (see Chapter 5) to enable them to explore non-functional requirements and ensure that they are identified at an early stage rather than being considered as a later (and often more expensive) addition. In a competitive business world, organisations cannot afford such legal transgressions and business mistakes. It is often the work of the business analyst that analyses the needs and ensures the solution includes the processes and systems to overcome them.

Hierarchy of requirements

Requirements are related to each other. Some of the general and technical requirements are elaborated and expanded in the non-functional and functional requirements; some functional requirements are related to non-functional requirements. All of the requirements are driven by the organisation's values, strategy and objectives. If a requirement is not aligned with these then it needs to be explored further as it is questionable that it is really a requirement. Understanding the hierarchy of requirements helps ensure that the requirements are consistent and coherent. When we define a functional requirement the business context provides the basis that underpins and supports the requirement. When eliciting non-functional requirements, the business and technical policies help give insights into why these requirements are necessary at the level of performance stated.

For example, data protection legislation defines the principles to be adopted by any organisation that stores personal data. This requirement will be elaborated further in the non-functional requirements to define the security levels required for specific sets of data, and the requirements concerning access restrictions, backup and recovery. Further, the functional requirements that define the data requirements will be linked to the security and legal requirements for this area.

The hierarchy of requirements, linking functional and non-functional requirements back to the general and technical business requirements, provides a means of tracing the original business need for the requirements. This helps when considering the priority of the requirements, the timescale for delivery and the possibility of removing a requirement.

Documenting a requirement

Each requirement is documented to define clearly what is required. A requirements catalogue template is provided below showing the range of characteristics that may be

documented about an individual requirement. An example requirements catalogue entry is shown in Appendix 11A. The exact set of characteristics used to define requirements will depend upon the organisational standards and development approach to be adopted.

Requirement identifier	The unique identifier allocated to the requirement. This is often a code that is linked to the type of requirement. For example, the technical requirements may be allocated the identifier T-n such as T-001, T-002, etc. The identifier may also include a version number, including a draft version number for when the requirement is still to be reviewed and agreed. An identifier may be: G-006v0-1 to indicate a general requirement in its first draft version; F-028v2-0 to indicate a functional requirement in its second reviewed and agreed version. Alternatively, the version history for the requirement may include the version number (see below).
Requirement name	The name allocated to a requirement. This is a short descriptive phrase that indicates what the requirement concerns.
Requirement description	The description should provide a clear definition of the requirement. Initially, the description may be at an outline level and elaborated in more detailed versions of the requirement documentation. When describing requirements it is good practice to adopt the following structure: <ul style="list-style-type: none"> • actor (or user role) • verb phrase • object (noun or noun phrase). An example functional requirement is: ' <i>the receptionist shall be able to view the customer name, address and telephone number</i> '.
Source	An example general requirement is: ' <i>the solution shall comply with the provisions of the Data Protection Act</i> '.
Owner	The originating person or information source for the requirement. This may be a stakeholder or could be a document containing information relevant to the project. For example, a stakeholder may have identified the requirement during an interview or other discussion, or there may be an earlier document – such as a project brief or feasibility study – that includes some of the business requirements.

responsible for the business function or department, who has the authority to approve the definition of the requirement.

Author The analyst who has elicited and documented the requirement.

Type of requirement The categorisation of the requirement. It may be sufficient to indicate whether the requirement is general, technical, functional or non-functional – although it may not be necessary to state this if the identifier includes a reference to the type. The type of requirement may be defined at sub-category level, for example, a requirement may be general, legal.

Priority The level of priority of the requirement. The approach used here varies between organisations but can also vary between different projects within an organisation. Sometimes a straightforward approach of high, medium or low priority is used with the organisation deciding the implications of each level. Similarly, sometimes the categories 'mandatory', 'desirable' and 'nice to have' are used. The Dynamic Systems Development Method (DSDM) defined a richer approach to prioritisation using the mnemonic MoSCoW. This approach is particularly suitable where several increments of a business change solution are to be implemented or iterative development is to be used to develop a software solution. The MoSCoW mnemonic stands for:

M – Must have. Mandatory in the first increment. Absolutely essential.

S – Should have. Mandatory but may be deferred (for a short period) to the second increment.

C – Could have. Desirable but may not be implemented due to time and budget.

W – Want to have but won't have this time. Identified as a requirement to be deferred until a later stage. There may be several reasons why a requirement is deferred. Some requirements are recognised as needing further consideration and would cause delays to some of the mandatory requirements if they were to be implemented in the first increment. Other requirements may require a later implementation for business reasons. This may be because it concerns an element of the business strategy that is due to be put into operation at a later point or could be an anticipated legal change.

Business area The name of the business area to which the requirement belongs. This may be the name of the business function or department. Alternatively, a more detailed approach may be useful and the name of the business process or use case may be used.

Stakeholders	The job roles or names of any stakeholders with a particular interest in the successful resolution of this requirement, and the details of their interest. Identifying stakeholders and their interests for each requirement provides a useful prompt to the business analyst to ensure that all relevant stakeholders' interests have been considered.
Associated non-functional requirements	Some functional requirements are associated with specific non-functional requirement. For example, there may be a business customer service policy that guarantees a speed of response to information requests. As a result, the functional requirement about accessing customer account information may have a performance response time non-functional requirement associated with it. An alternative approach is to document related requirements (described below).
Acceptance criteria	The criteria that will enable the business staff to formally agree that the requirement has been delivered. For each requirement, we should consider how we can check or measure if a requirement has been met.
Related requirements	The identifiers of any requirements that are related to this requirement. They may be related for several reasons: there is a higher-level business requirement that provides further business information or justification for a functional or non-functional requirement; there are non-functional requirements concerning areas such as usability or security that affect functional requirements or vice versa; there are other requirements that concern a similar general, technical, functional or non-functional area. The identifier for each of the related requirements should be listed here.
Related documents	The identifiers for any documents that provide further information about this requirement. These documents may be project documentation such as the project initiation document or may be business justification documents such as the business case. Another form of documentation that may be linked to the requirements is the modelling documents that have been created for the business change project. Some of these models may be contained within the requirements document however, it is still useful to show where there are requirements that are related to them.
Comments	Additional comments that the analyst feels are useful to document for a particular requirement.
Rationale	The business justification for the requirement. The rationale for a requirement may be cross-referenced to specific benefits in the business case.

Resolution	The outcome of a requirement. There are several possibilities as a requirement may be implemented, deferred for consideration in a later increment, merged with another requirement or dropped. The resolution will be used to record the decision and the timing of this decision.
Version history	The history of the requirement through the different versions that have been created. This information should include the version number (although as discussed earlier this may be combined with the identifier) and the date. Each version should also record the reason for the change to the requirement and the reference to the change control documentation.

However, producing a full definition for each requirement will be extremely time-consuming and could be wasted effort in some situations. The level of detail of the definition will depend upon several factors:

- The stage of the analysis, is it an initial view of the requirements or a more detailed requirements specification?
- The nature of the solution, for example, a business process change or IT system replacement.
- The level of priority of each requirement. This is an essential piece of information and will help to prioritise the requirements work. For example, if a requirement is allocated a W priority, the detailed work should be deferred until the point where it is to be included in the solution.
- The approach to be adopted to deliver the solution, for example, evolutionary system development or off-the-shelf software purchase.

Some aspects of a requirements catalogue definition will emerge earlier than others; initially, we may only document the identifier, name, description, source and author. However, once more detailed requirements analysis has been performed additional aspects such as owner and priority will be defined. After the requirements catalogue has been structured and duplicate or overlapping requirements removed, features such as the related requirements will be stated. Cross-referencing to other documents or models may be done late in requirements analysis. The resolution of a requirement may only be entered once the requirements have been validated and this may be subject to change if a MoSCoW prioritisation approach is used. The version history will only be required if the requirement changes.

The requirements catalogue is a central document throughout a business change project. It records what is required, the business justifications, sources of information and a rich network of connections. The level of the descriptions needs to be sufficient for the purpose rather than over-engineered for any eventuality. Sometimes the business stakeholders are unable to provide the precise requirements in extensive detail and the approach that will deliver the requirements needs to take account of this. In this situation, the description still needs to be clear about what is required but may not contain the complete set of details; a useful distinction is to separate what is required from how it will be delivered. Sometimes, the business staff have decided what is

needed but some of the finer detail concerning how the requirement will ultimately be delivered may still be open for discussion. In this situation, the requirements catalogue should be clear about what is required and leave the further detail to be explored using approaches such as scenarios and prototyping.

MANAGING REQUIREMENTS

A failure to understand, document and manage requirements often lies at the heart of problems with business and IT system changes projects. While a structured, well defined set of requirements will provide an excellent basis for a change project, problems can still occur if the requirements are not traceable. The traceability of the requirements is a critical quality characteristic. There are two forms of traceability: horizontal and vertical.

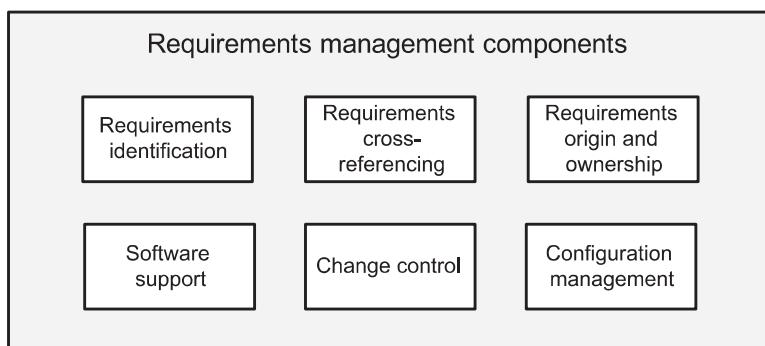
Horizontal traceability concerns tracing the requirement from inception to delivery. We can think about 'backwards from' and 'forwards to' horizontal traceability:

- 'Backwards from' traceability involves the ability to trace the source of a requirement from any later point in the business change or software development lifecycle. It answers the question 'what was the source requirement for this feature of the solution and who raised it?' We need to be able to identify where a requirement originated so that we can seek clarification from the source where necessary. This is particularly important when requirements are in conflict or there are conflicting views as to the priority of a requirement.
- 'Forwards to' traceability involves the ability to identify any requirement and track where it has been further developed and ultimately implemented. It answers the question 'what happened to this requirement?' and should show that each requirement has been resolved satisfactorily.

Vertical traceability concerns tracing a requirement up or down the hierarchy so answers the questions about alignment with business values, strategy and objectives.

Requirements management is essential if there is to be traceability of requirements. There are several elements involved in managing requirements as shown in Figure 11.4 and described below.

Figure 11.4 Elements of requirements management



Requirements identification

Each requirement needs to be identified uniquely in order that any reference to that requirement corresponds to only one requirement.

Cross-referencing

All related requirements and documents should be cross-referenced so that further elaboration or information concerning the requirement can be accessed easily. During requirements management, the cross-references provide the basis for the impact analysis of proposed changes. They allow the analyst to identify which requirements are related to the requirement that is the subject of the change, and consider whether the change will affect the other requirements.

Origin and ownership

The source identifies the origin of the requirement. This is the person or document the requirement should be traced back to during 'backwards from' traceability. This is the origin of the requirement and, whether this was a person or a document, will be able to provide additional information and justification about the requirement. When considering changes to the requirement, the source can help to clarify the impact of the change and as a result help with the decision about the change.

The owner typically has responsibility for the business area affected by the requirement and will need to make decisions about the requirement. It is important therefore that this person is involved as the change project unfolds. The owner will be a key stakeholder should any proposed changes arise that will affect the requirement, either directly or indirectly via related requirements. In addition, fundamental business changes such as budget reductions or timescale changes may require requirements to be discontinued, reprioritised or delayed; the owner will be the senior person in the decision-making process for the requirement and will have the ultimate authority concerning the requirement.

Configuration management

Configuration management is concerned with controlling any changes made to project deliverables, such as documents, in order to ensure that the changes are made in a disciplined manner and traceability sustained. Without effective configuration management the requirements document can become inaccurate as the following problems can occur:

- inability or difficulty in identifying the latest version of a requirement;
- the reintroduction of out-of-date requirements;
- using the wrong set of requirements for further development or testing work.

As a result, it is important that appropriate mechanisms are developed that will manage the implementation of any changes to the requirements. There are two key areas to consider:

- configuration identification;
- configuration control.

Configuration identification

Configuration identification is concerned with defining the following:

- The deliverables to be brought under configuration control. These are known as the configuration items (CIs). During requirements management, the deliverables will include the individual requirements catalogue entries, the composite set of entries that form the requirements catalogue, the models that elaborate and define the requirements, the requirements document. There should also be a structure showing how these configuration items relate to each other.
- The identifier and version numbering scheme to be applied to the CIs. As discussed previously, each requirement has a unique identifier but if we are to control the other configuration items then this will also be required for each of them. For example, the requirements catalogue will require an identifier. Where a CI has been identified, and has an identifier, it will also require a version number. The approach to be adopted for allocating version numbers will need to be defined. Version numbers will apply to each individual CI. For example, while a requirement will have a version number, so will the whole requirements catalogue within which the requirement sits.

Configuration control

There will also need to be a process for controlling the CIs and ensuring that they are not changed without formal approval and version numbering. This is typically called a version control process. There are several elements to this:

First, a CI is created in draft form. While the item will have an identifier, the version number will need to indicate that the CI is in draft form. One approach is to number initial drafts using the format 0.n. For example, a draft requirement catalogue entry with an identifier T-007 could be numbered T-007v0.3 indicating that this is the third draft version of the technical requirement numbered 007.

Second, as described in Chapter 10, requirements validation is carried out to review and agree the requirements documentation. At this point, the CIs that describe the requirements are said to be 'baselined'. This means that they are brought under configuration control and cannot be changed by anyone without following the formal configuration management procedure. When a CI is brought under configuration control it is allocated a version number reflecting its baselined status. Using our example above, the technical requirement would have the identifier/version number T-007v1.0. Sometimes it can be useful to baseline a requirements document prior to requirements validation, when rigorous requirements analysis has been carried out and a formal approach to change control is felt to be beneficial.

Third, once a CI is baselined, no changes can be made to the content without the approval of the configuration manager. All of the CIs should be stored in a secure area so that they cannot be accessed and revised at will. If a change to an item is approved, the configuration manager releases it for revision. Once the item has been amended and the revised version brought under configuration control, the new version is renumbered. In our example, the requirement would now be allocated T-007v2.0.

Configuration management in an Agile environment

Configuration management needs particular consideration when an Agile development approach such as DSDM or Scrum is used. Because these approaches tend to embrace change and explore requirements using prototyping approaches, initially much of the information about the requirements will be defined within the prototypes rather than the requirements documentation.

While the high-level requirements documentation should be relatively stable, the prototypes that are used to elaborate and implement the requirements are likely to change regularly. Having baselined the requirements document reasonably early in the development project, this document will provide an audit trail for the high-level requirements. Further detail regarding the requirements may be added at a later stage. Baselining and controlling the versions of the prototypes will also be important if the requirements are to be managed.

There are several possibilities for baselining the prototypes during Agile software development including:

- **Baselining every prototype before demonstration:** this has the virtue of clarifying the version that has been demonstrated to the business users.
- **Baselining daily:** this is highly disciplined but can prove onerous and unnecessary.
- **Baselining at the end of a timebox:** this is fine if the timeboxes are reasonably short – a few days perhaps – but less sensible with longer timeboxes or, for example, the 21-day ‘sprints’ used within Scrum.

Each baselined prototype is placed under configuration control as this is the only way to ensure that the up-to-date version of the prototype is used. The CI will consist of the actual prototype, the tests run on it and the record of the users’ comments. Thus, as the prototypes are developed and refined, a complete audit trail is created of the changes made and – very importantly – why they were made.

Change control

Changes occur frequently on projects. This may be because of external factors such as legal or regulatory changes or competitive forces, or may result from internal changes, such as strategies, policies or people. As a result, any requirement may be subject to change during a project. Having accepted that changes will happen, then requirements management should include a process to handle these changes. The stages of this process are as follows:

- **Documenting the proposed change.** The change should be documented as a ‘change request’, stating who raised the change, a description of the change and a justification for requesting it.
- **Consulting the stakeholders.** Each change request should be sent to representative stakeholders to assess the impact of the proposed change, including the effort to make the change and the corresponding cost.

- **Deciding on the change.** The change request and the impact assessment should be reviewed by the designated approval authority. If the change is approved for implementation, the CI is released by the configuration librarian so that the change can be applied and the new version created. The combination of the configuration management and change control approaches provide a means of creating a version history for the requirements documentation. Each change that has been applied to the requirements is documented and explains why one version was changed to create the new version. Over time, a complete audit trail will be created explaining what actions were taken, why this was done and when.

Software support

Most requirements documents contain too many requirements to manage the set of documentation, cross-references and versions manually so an automated tool is usually required. These tools should provide the following features:

- **Documentation creation and storage.** This will require editing tools that provide facilities such as word processing and modelling. Further, some tools provide document management capability that incorporates functions such as publishing documentation for access by authorised stakeholders, allowing on-line reviewing and tracking revisions.
- **Secure storage and access.** If the documentation is to be placed under configuration control, it will be essential for there to be restricted access to the individual documents.
- **Documentation linkage.** The cross-references between documents will need to be recorded and related documents accessed easily. This will help in requirements management activities such as impact analysis and tracing version histories.
- **Version numbering.** The allocation of identifiers and version numbers to configuration items.

Some specialist toolsets provide a range of features, including those above, designed to support rigorous and efficient requirements management. Some may be integrated tools providing functionality that support later activities such as code generation and testing. However, many organisations use automated tools that are not designed for requirements management specifically and may offer just generic features such as diagram creation and word processing. Such software can offer the key functionality required for requirements management but will also require a great deal of manual activity to supplement the automated support.

SUMMARY

Well defined and traceable requirements are undeniably key to a successful business change project. They set out what a solution needs to deliver and the relative priorities of the features to be included. The requirements document needs to be structured clearly so that a thorough requirements validation can be conducted by the reviewers. Traceability of the requirements is also critical if the business is to be able to confirm that their needs have been met and all decisions regarding the requirements are to be transparent and auditable.

APPENDIX 11A

REQUIREMENTS CATALOGUE ENTRY							
Project ID and name: Trent Cars Sales Improvement Project							
Author:	Date:	Version:	Status:	Page:	of 1 1		
J Williams	02/09/14	0.1	In development				
Requirement ID	G-001v0.1						
Requirement name	Compliance with Data Protection Act						
Business area/domain	Sales, Servicing, Customer Services						
Source	L. Stevens, Customer Services Manager						
Owner	W. Brown, Managing Director						
Priority	M						
Type of requirement	General						
Requirement description	The solution shall comply with the eight principles of the UK Data Protection Act, 1998						
Associated non-functional requirements	See related requirements						
Acceptance criteria	A user without authorised access to personal data shall be advised that they do not have the authority to access the data and shall not be allowed access to the data						
Justification	Legal requirement. Non-compliance could result in the receipt of an enforcement notice from the Information Commissioner's Office or the Information Commissioner's Office could impose a financial penalty on Trent Cars						
Comments							
Related documents	Memo from W. Brown 'Data Protection Act compliance' dated 12/06/08. Information Commissioner's Office website www.ico.gov.uk						
Related requirements	N-005 Data Access Restrictions; N-110 Data Security						
Resolution	To be implemented in phase 1						

FURTHER READING

Cadle, J., Paul, D. and Turner, P. (2014) *Business Analysis Techniques: 99 essential tools for success*, 2nd edn. BCS, Swindon.

Robertson, S. and Robertson, J. (2012) *Mastering the Requirements Process*, 3rd edn. Addison Wesley, Harlow.