Chapter 10

Requirements Specification

Source:
STARTS
Purchasers' Handbook
Second Edition 1989

Prepared by the Industry with the support of the DTI and NCC

CONTENTS

1 Introduction

Chapter 3 gave an overview of the life cycle and indicated the need for a disciplined framework for software development. It also highlighted the need for explicit agreement to be reached on the various responsibilities of both the purchaser and the supplier during contract execution.

This chapter gives guidance on "good practice" in a critical area where purchasers and suppliers need to communicate - Requirements Specification (RS). Indeed the RS may well form the basis of the contract between purchaser and supplier, defining what the supplier is to build and what the purchaser is willing to accept.

Whilst the task of preparing a requirements specification for a realtime system is difficult, it is also important. One difficulty is that, in general, it is impossible to know, at the requirements stage, what functionality will be implemented in hardware and what will be implemented in software. Consequently, it will be necessary for the RS to cover the complete system, not just the software. This discussion of the RS assumes that it has this wider scope.

The advice on requirements specification given in this Handbook is in three parts:

* this chapter which summarises the problems of requirements definition and outlines the recommended content of a requirements specification

* Appendix B which gives a detailed checklist and description of the recommended content of a requirements specification and guidance on the activities of requirements definition

* a set of checklists, at the end of this chapter, which identify the contents of the requirements specification and draw out key points about the requirements definition process.

The checklists are not intended to stand alone but rather to be an aidememoire after the chapter (or the chapter and the appendix) have been read and understood.

1.1 The Importance of the Requirements Specification

The exact role of the RS in a procurement depends upon the particular procurement strategy employed. However, the RS always fulfils two roles:

* the primary input to the design process

* the baseline against which acceptance tests are carried out. Thus it has a pivotal role in any major software system.

Perhaps unsurprisingly, experience has shown that purchasers who fail to provide their chosen supplier with an adequate Requirements Specification are likely to find that the supplied system does not adequately meet their needs. Furthermore, as has been shown in

chapter 2, the costs of rectifying system deficiencies after installation are likely to be much greater than the cost of preparing an effective RS in the first place, if indeed rectification is practical at all. Any corrective action will also result in delay to the successful utilisation of the system. Thus the preparation of a good RS reduces both cost and risk.

For these reasons, the importance of preparing an adequate Requirements Specification is now becoming appreciated. Other organisations have, of course, recognised these problems and, as a result, some of them have published guidelines to assist users in the task of preparing specifications. Some relevant publications are given in figure 1.

1.2 The Difficulty of Preparing an Requirements Specification

The RS must define every aspect of the way in which the system is to perform including behaviour in error conditions. Account must also be taken of environmental factors, the expected lifetime of the system and possible changes to the requirements during the system's life.

The scope of an RS makes its production a difficult task but the problems are compounded by its role in the system life cycle. The RS is:

* the major interface between the purchaser and supplier

* the first technical task in the life cycle

* concerned with documenting hitherto unstated requirements.

The differing cultural and technical backgrounds of purchaser and supplier lead to differing interpretations of the RS leading, inevitably, to communication problems between them.

Further, the RS itself has to satisfy conflicting requirements - it must be precise and also intelligible, or understandable, to both purchaser and supplier.

The RS may also be used as a basis for a contract and thus be subject to the constraint that it should be legally, as well as technically, meaningful. This means that, should a disagreement occur over the completion of a contract, it should be possible to determine, unambiguously and (if necessary) in a court of law, whether or not the supplier has fulfilled the contract.

These primary problems of producing an RS are caused by its role in the life cycle. More detailed, technical, problems are discussed in section 4.

These, and other, difficulties mean that it is very desirable to have the support of good methods and tools at the requirements stage.

1.3 Terminology

The roles and activities relevant to the requirements stage are rather different from those at subsequent stages. It is therefore helpful to define terms:

* Requirements Specification (RS) the product of the requirements stage (see section 6 in this chapter and Appendix B for a definition of its contents)

* Requirements Definition (RD) - the complete process of producing the RS.

1.3.1 Activities of RD

* elicitation - capture of requirements

* verification - checking internal consistency and completeness of an RS

* validation - checking that the RS reflects the purchaser's needs.

1.3.2 Roles

* Purchaser - the organisation which is responsible for the procurement of the system and thus for ensuring adequacy of the RS.

* Supplier - the organisation which is responsible for implementing the system in accordance with the RS.

* Users - those who will use the implemented systems to perform their work; the main source of information for the RS; typically, but not always, part of the purchaser's organisation.

* Analysts - people whose task it is to produce the RS; they may be purchaser's staff or consultants.

Note that these are genuine roles and one individual may fill more than one role.

In many cases, the system to be constructed will interact with other equipment, not a person. If so, the equipment designer will be the main source of information about interaction with the equipment. To avoid circumlocution, the term "user" is taken to cover equipment designer, genuine users and their representatives.

1.3.3 Component

There is some terminological difficulty when considering the structure of a system and its operational environment (the larger system in which it works). Here, the term "component" is used for all units within the system and its environment. Thus a functional component might be thought of as a subsystem. An environmental component will typically be a user or equipment with which the system interacts. This term is not meant to convey any connotation

of design or scale - indeed an environmental component might be an individual or a global telecommunications network.

2 Purpose and Scope of this Chapter

An ideal RS would contain everything the supplier needs to know in order to produce a system which is acceptable to the purchaser and nothing more. Even given this clear statement of objective, it is not obvious how to produce a good RS nor even what constitutes a good RS. The purpose of this chapter is to throw some light on this issue.

2.1 Aim of the Guidelines

The aim of the guidelines given in this Handbook is to provide purchasers' project managers with advice on key facets of the procurement of major software based systems.

The specific aim of this chapter, together with Appendix B, is to:

* identify the problems of RD
* give a checklist for the contents of an RS
* give guidelines on how to carry out requirements capture and analysis under different procurement conditions
* provide advice on the suitability of structured and formal specification methods and tools.

A subsidiary aim is to give suppliers a clearer idea of the form and content they should expect of a purchaser's RS.

2.2 Relationship with Other Guidelines

Although there are a number of other relevant guidelines, it has been necessary to produce a separate guideline because the others have a different coverage. The IEE and IEEE guidelines deal only with software requirements and the BSI standard is very general. Nonetheless, these other guidelines (see the opposite page) contain much relevant material and readers are therefore advised to consult the documents identified.

2.3 Relationship to Other Chapters

This chapter is related to those on project management (Chapter 4), Safety Related Systems - SRS - (Chapter 7), Procurement Strategies (Chapter 9) and Product Acceptance (Chapter 12). The relationships between Chapter 7 and 9 and this chapter are of sufficient importance to warrant special attention.

Safety criticality or otherwise of a system affects the *content* of the RS. Here, where information on aspects of systems safety should be included in the RS is indicated but

treatment of detail is left to the SRS chapter.

Procurement strategy significantly affects the *process of producing* the RS. This chapter expands on the role of requirements in different procurement strategies but discusses the content of the requirement specification and the principles of requirements definition independently of the procurement strategy.

(The full STARTS Purchasers' Handbook is available viewing - see lecturer)

2.4 Relationship to the STARTS Guide

The STARTS Guide and Handbook are compatible even though, due to the time between publications, there are differences in detail and emphasis. Most significantly, the Guide has a three level checklist for the contents of an RS whilst that in this version of the Handbook has only two levels. The change in structure has been made to improve the clarity of the checklists. No points have been lost-indeed those in this Handbook are more extensive than those in the Guide.

PUBLISHED GUIDELINES TO ASSIST IN THE PREPARATION OF SPECIFICATIONS

Guidelines for the Documentation of Software in Industrial Computer Systems

IEE 1985; these guidelines are slanted to wards software for smaller industrial process control applications. They cover the requirements for documentation at the main stages of purchasing such systems. The sections on User Requirements Specification, Functional Specification and Software System Specification are particularly relevant.

IEEE Guide to Software Requirements Specifications, ANSI 1984; this deals only with specifying software requirements but covers a wide range of applications. It provides very useful detail of good practice.

British Standard Guide to Specifying User Requirements for a Computer System British Standards Institution BS 6719 1986; this guide again covers a wide range of applications. Its approach is rather different from the IEE guidelines.

Figure 1

3 The Role of the RS in the Procurement Process

A primary management concern is to reduce the risk of the procurement process. The choice of procurement strategy, how to produce the RS and when to move into full development are key factors in this risk management process.

There are at least three distinct approaches to procurement strategy: contractual, evolutionary and prototyping (see below). The first two strategies are also known as staged and incremental. In each case, the RS retains its central role as a baseline for development and acceptance testing but details of the processes differ.

Chapter 9, Procurement Strategies, discusses these approaches in detail. Two of the other three "strategies" described in Chapter 9 correct resourcing and generic systems - are also

applicable. The third "strategy" - technical approach - includes prototyping.

It is instructive to discuss the stages prior to production of the RS, Requirements Definition and the subsequent development activities.

TYPES OF PROCUREMENT STRATEGY

Contractual	a full RS is produced prior to development.
Evolutionary	a partial RS (for a subset of the total system) is produced prior to initial development; the further functions are analysed and added to the RS prior to their development.
Prototyping	a two-stage process where the RS is initially imprecise but is refined into an RS suitable for a contractual procurement with the aid of a prototype. The prototype may

Figure 2

3.1 Pre-Definition

One aim in producing an RS is to reduce the risk associated with the development to an acceptable level. However, it is often necessary to carry out substantial activities prior to producing the RS as a further part of the risk management activities. (Note that this stage is discussed as part of RD in section 5).

Example scenarios are:

* Production of a completely new system for an identifiable set of users - here an outline RS and a prototype may be produced in order to clarify and validate requirements.

* Production of a completely new system for a perceived market, rather than specific users - here the main activities may be market and feasibility analyses.

* Replacement of an existing system with only minor functional change - here it will probably be satisfactory to proceed straight to the RS without any preliminary stages.

In each case, the decision points are different as are the scope for involvement by potential suppliers and the input to the RS. In all cases, however, it will be necessary to identify the key objectives or purpose of the system and make the decision whether or not to proceed to the full RS. Further, it is likely that some feasibility studies will be appropriate either in the pre-definition stage or during requirements definition.

3.2 Requirement Definition

The nature of the requirements definition process and the choice of analysis method depends on the procurement scenario and strategy. In addition, the primary source of information for the RS, in other words the input to the RD, is also dependent on the procurement strategy. In the three cases described on page 10.10, it will be, correspondingly:

* the user community

* the market and feasibility studies

* the existing system, its documentation and its operational environment.

The purchaser needs to use a method which is effective for requirements definition given the available information. In all cases, useful guidance on feasibility and other aspects of the proposed system and procurement can be gained through early discussions with potential suppliers.

The work of these early stages is the purchaser's responsibility, although the purchaser may have external assistance. For example, if the user organisation does not have the necessary skills to define the full requirement, a supplier or consultant may be invited to undertake the work. However, in order to avoid conflicts of interest, it is important that such assistance is kept distinct from the work of the eventual supplier.

For a contractual approach to procurement, it is recommended that, in order to increase confidence that requirements can be satisfied, a feasibility study is carried out before the detailed requirements specification is completed.

In other cases, for example, evolutionary development, design starts prior to completion of the RS. However, it is recommended that the RS is completed by an independent team (not the main supplier). This is again to avoid conflicts of interest. Clearly, close liaison is required between supplier, purchaser and analyst and these are key interfaces for project managerial attention.

There may be some circumstances where this advice cannot be followed. For example, the selected supplier may have some proprietary technology or specialist expertise which will affect requirements in a way for which an independent team could not cater. Conflicts of interest should still, however, be avoided and, in such circumstances, it is recommended that an independent requirements review team is used.

3.3 Development

As stated earlier, it is important that the requirements specification be used as the baseline for the subsequent stages of development and acceptance testing. However, as work proceeds, the requirement may change even if only to cover points which are exposed by the development process. It is therefore essential that the specification be kept up-to-date, particularly during the development and testing phases and, to do this effectively, disciplined configuration control procedures are required.

This requirement may seem bureaucratic when the evolutionary or rapid prototyping approach is employed. However, it is even more important in these circumstances. An objective of this phase is to produce a detailed RS leading to well documented and maintainable systems.

4 Problems in Requirements Definition

We have already described some of the reasons why producing an RS is difficult (see section 1.2). Some of the technical problems are inherent in the Requirements Definition process itself whilst others arise in determining the boundary between requirements and design.

4.1 Inherent Problems

Possible communication problems between purchaser and supplier were mentioned earlier. During RD, we are trying to document hitherto unstated requirements and similar communication problems can arise between analysts and users with different technical and cultural backgrounds.

Inherent problems include those listed on the opposite page. It is important that purchasers satisfy themselves that any methods and tools used in conjunction with reviews and other quality assurance procedures address these problems. Approaches to solving these problems are discussed in Appendix B.

4.2 Requirements and Design

These guidelines do not draw a sharp distinction between what the lEE guidelines refer to as a User Requirements Specification and the subsequent Detailed Requirements Specification although these correspond roughly to our level I and II specifications (see section 6).

We are, however, concerned with making a distinction between requirements and design, that is, between *what* a system should do and *how* it should do it. Requirements can be thought of as an external view of a system whereas the design represents an internal view.

INHERENT PROBLEMS in REQUIREMENTS DEFINITION

* Definition of the system boundary; it may not be clear where the boundary should be, that is, what functionality is performed within the system and what is performed without.

* Different users often have different, and sometimes incompatible, views of what the system should do and it is necessary to identify and reconcile these differences of opinion.

* Requirements are always subject to change due both to changes in the operational environment and in understanding as the system is implemented.

* Stated requirements are often untestable, for example, "the system shall be user-friendly"; it is necessary to re-express such statements in a form which is, in principle, testable or demonstrable.

* There is a tendency for information which is "obvious" to the purchaser to be omitted.

* Purchasers and suppliers often use different terminology leading to communications problems.

* Purchasers are frequently uncertain about the capabilities and limitations of computer systems and thus can state inappropriate requirements.

* Purchasers can have their own ideas on design which may be put in the RS.

* Users' expectations are often unrealistic and it is necessary to agree an acceptable and economically feasible compromise; users may also state attractive but inessential features and these should be eliminated as far as possible.

* In contrast, users' initial expectations can be less than the full requirements and it is necessary to analyse the whole scope of the problem to avoid a constantly changing RS as realisation dawns.

* The size of the RS should be proportional to the size of the underlying need; it is all too easy to insert an excess of detail on a well understood but minor area whilst almost ignoring key cost and risk areas when these are ill understood.

Figure 3

For a number of reasons, it is desirable to minimise the design content of an RS:

* The more design information there is in an RS, the further it is from users' experience and understanding and the more difficult it is to validate.

* Including design information in the RS reduces the suppliers' design freedom. This, in turn, can increase the cost of development as some cost-effective solutions may be ruled out.

* Design information in the RS can make it more difficult to adapt the system to (genuine) changes in requirements and thus the system becomes obsolete more quickly.

In practice, the distinction between requirements and design might be somewhat arbitrary. Indeed the requirements are usually dependent on what can be offered and are adjusted to suit

practical design needs. Further, it is often difficult in, for example, interface definition, to make a clear distinction between clarification of requirements and high level design.

However, some general guidelines on the distinction between requirements and design are given below. Although these are only guidelines, not hard rules, they will often help in deciding what information to include and what to leave out when producing an RS.

GENERAL GUIDELINES for the DISTINCTION BETWEEN REQUIREMENTS AND DESIGN

* The RS should contain the statement of requirements agreed between analyst and user. It may contain design constraints placed by the user but care should be taken to ensure that these are essential constraints. It should not contain constraints suggested by the analyst.
* Whilst it might be thought that an ideal RS would relate inputs to outputs and say nothing about the structure of the system, this can make the RS obscure. In practice, identifying one or two levels of function in the system and decomposing basic data structures to one or two levels probably gives the right balance between purity and clarity. It should be noted, however, that there is no reason why the design structure should be the same as the decomposition provided in the RS.
* Both users and analysts can provide information on requirements.

Figure 4

5 The Requirements Definition Process

Some aspects of the definition process have been dealt with in section 3 and, in addition, particular methods define their own process. Thus the discussion here is confined to aspects of the process which are independent of procurement strategy and the requirements definition method.

5.1 Process Overview

It is recommended that requirements definition be carried out in two main phases which correspond with the activities identified in sections 3.1 and 3.2, that is:

* Early discussion within the organisation and with potential suppliers or consultants to establish the crux of the requirements.

* Preparation, in stages, of a two level document setting out the requirements:

Level I an introduction to the requirement giving:

a) sufficient background for the supplier to be able to understand the role the system is expected to play;

b) a description of the requirement from the viewpoint of the users.

Level II a fully detailed specification of the requirements which should:

a) provide enough information to enable the development to proceed without substantial further recourse to the users;

b) be a suitable basis for competitive tendering.

Level I corresponds to what the IEE refers to as the User Requirements Specification and may be thought of as a "Cardinal Points" specification.

5.2 Early Discussion Phase

Interaction with consultants and potential suppliers helps purchasers to assess a number of points, for example:

* The detail which needs to be included in the RS and areas where the purchasers will need help in preparing the specification.

* Whether the purchasers have made implicit design assumptions which are unwarranted, for example, a potential supplier might be able to offer a better solution to their needs, perhaps by using new technological developments or alternative designs.

Certain individuals/roles must be identified:

* a set of users whose views are to be considered

* authority for approving the RS and resolving conflicts between users.

In addition, this phase helps purchasers identify which functions, etc. are essential as distinct from those which would be desirable if they could be provided at reasonable cost. This assists in the development of a procurement plan.

The early discussion phase may be conducted on an informal basis. The objective is to help purchasers take the right path in the subsequent stages.

As a result of interactions during this phase, the user organisation may rethink its needs. It is crucially important, however, that changes made to the outline plan, at this stage, are in accordance with the users' real needs, both short and long term, and include all aspects of the expected use of the system, for example: corporate plans, training needs and possible changes of operation.

A major outcome of this stage is a decision whether or not to proceed with developing the full RS.

5.3 Production of the Full Requirements Specification

Since Requirements Definition is discussed independently of procurement strategy, this section covers initial development of the RS and changes to it, regardless of the source from

which the request to change arises.

Thus there are three classes of input to the evolving RS:

* initial and continuing input from the users

* change requests arising from the development process and system use

* change requests arising from changes in the system environment or perception of the environment.

The balance between these inputs depends on the procurement strategy chosen and the stage reached in the project.

The process itself must deal with the problems identified in section 4.1. Note: There is no implication that there must be a stage of the method corresponding to each item listed.

A number of methods explicitly identify viewpoints of a system. Viewpoints studied should include the system's environment so that the system boundary can be identified. They should also cover development and maintenance so that support issues are properly addressed. When considering safety related systems, the viewpoints should be expanded to include the certification agency.

In addition, any requirements definition method must encompass:

* elicitation of requirements from the users

* verification of internal consistency and completeness of the RS

* validation of the RS.

Ideally, a method should include guidelines for each of these activities (see the STARTS Guide and Appendix B).

The preparation of the levels I and II documentation may be undertaken by the purchasing organisation alone or with help from a supplier or consultant. Normally the purchasing organisation will prepare level I, possibly after preliminary discussions with consultants or potential suppliers.

With level II, unless the purchaser has adequate skills, its preparation should be left largely to consultants or potential suppliers. However, purchasers must satisfy themselves that the final, level II, detailed specification is in accord with their requirements. The purchasing organisation should, therefore, work closely with whoever is preparing the detailed documents and employ staff who are competent to understand the specification and its implications.

6 Contents of the Requirements Specification

It is recommended that, within the two levels described in section 5.1, the Requirements Specification contains the sections listed below. The numbering of the sections within level II starts at 2 to avoid confusion with level I.

Although the distinction is not always clear, Part A defines what it to be produced whilst Part B covers constraints on the production process. Indeed, if the RS it used as a contractual document, it may be that Acceptance Requirements is the legally binding part of the document from the point of view of what the delivered system should be. Furthermore, the processes to be followed in carrying out quality assurance and project management are a vital part of the project requirements; for example, the only way of achieving certain levels of quality is via specifying the processes to be followed.

It is not suggested that the specification be produced as a single operation; it should be planned as a staged exercise with allowance made to iterate as necessary. This applies, in particular, to the Detailed Requirements Specification (Level II) which forms the basis for design, implementation, the change control process, quality assurance and acceptance testing.

LEVELS I and II of a REQUIREMENTS SPECIFICATION

OVERVIEW and OUTLINE REQUIREMENT	Level I	1. Overview and Outline Requirements
DETAILED REQUIREMENTS SPECIFICATION	Level II	
	Part A	2. System Overview and Environment
		3. Data Structure and Relationships
		4. Functional Component Specification
		5. Design Constraints
		6. Life Cycle Aspects
	Part B	7. Installation Aspects
		8. Deliverables
		9. Acceptance Requirements
		10. Project Management and Quality Assurance

Figure 5

6.1 Level I Specifications

The Level I specification should provide:

1. A succinct requirement specification for management purposes.

2. A statement of key objectives which might be called a cardinal points specification.

3. A description of the environment in which the system is to work.

4. Background information and references to other relevant material, for example,

corporate strategies and feasibility studies.

5. Information on major design constraints, for example, time scales, finance and inter-operability with other systems.

The expected content of each section is described in Appendix B. All subsequent requirements should be traceable back to Level I.

6.2 Level II Specification

The Detailed Requirements Specification should be set out using the headings and structure outlined above and given, in detail, in Appendix B, subject to this being compatible with the method used to prepare the specification. If alternative structures are used, the contents should cover the same points; in any event, it is essential to preserve the traceability between Levels I and II. In order to achieve this traceability, it is helpful to uniquely reference each requirements statement and to keep each one brief, for example, two or three sentences or one diagram.

Part A (sections 2 to 5) is concerned with requirements for the *product* of the development process, Part B (sections 6 to 10) with requirements and constraints on the development *process itself.*

6.2.1 Part A

Part A should define in detail:

* all the data input to, and output from, the system

* the behaviour of the main functional entities in the system's environment

* the functionality and attributes, for example, performance, of the main system components

* the design constraints, for example, standards and interfaces, placed on the system as a whole and on individual functional components.

Section 2 - Overview and Environment

Section 2 should give an overview of the system and specify the behaviour of the environment in detail. It is important to identify the environmental function since changes here must be traced to, and reflected in changes to, the system requirements.

The system overview should identify major aspects of the system (such as start-up and close-down, built-in test and operating modes) which may not easily be discernible from the detailed descriptions in section 4 - Component Specification.

The environmental specification should cover "components", that is, equipment, systems and

individuals (or roles), with which the target system directly interacts. As with all the component descriptions in the RS, this material should cover both functionality (what the component does) and non-functional requirements (constraints on how it does it). Physical aspects of the environment, for example, heat, light, power, should also be included here.

More precisely, the term "functional" is used to refer to requirements which can be expressed purely in terms of relationships between input and output values. The term non-functional is not pejorative and simply implies that the requirement cannot be stated purely in terms of data relationships.

These non-functional requirements are of three types:

* **Performance,** including speed of processing, data volumes, store usage

* **Dependability,** covering a range of constraints including safety, reliability, availability and security

* **Quality,** covering other important attributes such as ease of change, interoperability, intelligibility and ease of use.

The detailed set of issues to be covered will vary from system to system.

It is important to address these non-functional issues, even for the environmental components, as data rates and volumes and failure modes in the environment will affect the system functionality.

Section 3 - Data Structure and Relationships

Section 3:

* identifies all data items, their structure, relationships and attributes

* indicates which functional components use and generate each data item.

Some people take the view that the data held by the system is included in the internal workings of the system and therefore part of the design process rather than requirements definition. In practice, however, it is usually very difficult to specify the functions of a system without defining the data it holds and operates upon. Indeed, as a consequence of this, a number of methods now take the view that the data model is central to the whole process of requirements analysis.

Section 4 - Functional Component Specification

The most complex, and probably most important, part of the RS concerns the functionality of each component. Thus section 4 is the core of the RS; it itemises functionality, data dependencies, control mechanisms, failure mode behaviour and non-functional attributes including safety.

Functional component specifications are often highly complex in themselves. It is therefore important to structure them in such a way that the requirements can be stated and communicated as easily as possible and details of the recommended method of doing this are given in Appendix B.

Note that a functional component may, if convenient, be treated as a sub-system and defined in terms of a further level of components. In general, however, breakdown to more than two levels is not desirable as this is encroaching on the domain of design.

It is worth stressing here certain attributes which are often detrimentally omitted from requirements:

* **Interface protocols** which define the communications protocols for data which is passed to or from the system.

* **Fundamental assumptions** which indicate aspects of the requirements which are expected to be stable for the lifetime of the system; changes to these may lead to extensive rework, or even the discarding, of the system.

* **Expected changes** to the requirement, for example, changes in terminal types to be supported; the system should be designed to accommodate such changes.

Section 5 - Design Constraints

The design constraints which the purchaser requires to impose on the new system should be detailed in section 5 rather than section 4 so that there is a clear separation of concerns. Broadly, the constraints can be divided into three categories:

* Software constraints which include requirements for compatibility and inter-operability.

* Hardware where there is an unfortunate tendency for over-constraint, for example, by specifying a particular make of computer. There can, however, be legitimate constraints. For example, in an avionics application, power and weight considerations would affect the choice of hardware.

* Human constraints which include skill levels which can be expected of the system's operators.

6.2.2 Part B

Part B of the Detailed Requirements Specification should define:

* the system acceptance criteria

* installation procedures and constraints

* any constraints which the purchaser wishes to place on the supplier, for example, methods to be used.

Part B is discussed more briefly than Part A, both here and in Appendix B, as most of the material is dealt with in more detail elsewhere in the Handbook.

Section 6 - Life Cycle Aspects

This covers constraints such as the use of STARTS recommended methods and tools in the development process. It also addresses the longevity of the system, maintenance arrangements or requirements and likely needs for expansion.

Section 7 - Installation Aspects

These include conversion and training as well as hardware and software installation. It should be noted that conversion from an existing system to a new one may involve considerable work, for example, reformatting many megabytes of data, so that it may be necessary to develop special purpose conversion software.

Consideration should also be given to fallback positions, for example, parallel running of the old and the new systems until the new one is proven.

Section 8 - Deliverables

Everything that the purchaser requires to be delivered should be itemised and, if appropriate, the form in which it should be delivered. Items to be considered include program sources, documentation, software tools, training courses and hardware.

Section 9 - Acceptance Requirements

In addition to dealing with acceptance tests, this section should indicate the general conditions under which the system will be accepted. Thus topics such as phased delivery and periods of use prior to final acceptance should be considered.

Section 10 - Project Management and Quality Assurance

The project management and quality assurance requirements for the system development process should be documented here; these issues are fully covered in chapters 4 and 5.

7 Quality of the Requirements Specification

The quality of the RS is most important because any errors here are likely to go undiscovered until after the system is introduced into service. Discovery of flaws at such a late stage will result in the project incurring high costs and delayed timescales to correct the errors. It is therefore recommended that the RS be subject to thorough review (see section B.4. 1.5 in Appendix B).

7.1 Characteristics of a Good Requirements Specification

CHARACTERISTICS OF A GOOD REQUIREMENTS SPECIFICATION

1. **unambiguous -** every requirement should have only one interpretation. A glossary of terms can help but natural language has pitfalls and, to overcome this, formal and semi-formal languages should be used where practicable.

2. **complete -** all requirements and aspects of the environment, whether relating to functionality, performance, design constraints, etc., should be included. The RS should contain required responses to unintended events, as well as those that are intended. There should be no undeclared assumptions.

3. **testable or demonstrable -** used here in the sense that the requirement must be stated in such a way that the system can be checked to show that the requirement has been fulfilled. This is part of the Product Acceptance dealt with in Chapter 12, section 3.7

4. **consistent** - there should be no conflict between individual requirements.

5. **modifiable -** users usually alter their requirements on an almost continuous basis and the specification should therefore be organised in such a way that this can be accommodated, for example, with adequate structure and cross referencing. Note that redundancy may ease readability but present difficulties when changes are made.

6. **traceable - proper referencing** is needed to allow backward traceability (to previous documents) and forward traceability (to subsequent documents) -although clearly forward traceability cannot be checked when an RS is first produced. Thus individual requirement statements should be given unique references, for example, numbers.

7. **accurate and up-to-date during operation and maintenance pbase -** attention needs to be paid to operation and maintenance needs and the updating of the Requirements Specification to meet these especially where failure of a component could be critical.

8. **intelligible -** requirements have to be read by many groups of people - users, analysts, purchasers and suppliers. Consequently, perhaps the most important characteristics of a good RS is that it is intelligible to, and forms a basis for communication between, all the different classes of reader. Thus individual requirement statements should be very brief (perhaps *250* words) and focus on one topic only.

9. **modular -** it should be possible to produce, or understand, parts (modules) of the RS independently. This may mean that it is necessary to repeat some introductory material so that modules can be read on their own although this conflicts slightly with modifiability. However, it is essential that it is possible to

Figure 6

It is widely recognised (see, for example, the IEEE guidelines) that characteristics of a good RS include those given on the opposite page.

7.2 When is a Requirements Specification Complete?

Completeness is considered in detail since the question of "when to stop" is perhaps one of the most important to consider when producing an RS.

The difficulty can be simply shown by example: with a user interface, do you merely specify what information is to cross the interface or do you give the details, for example, data formats, command styles (menus, mice), etc.?

In some cases there may be a clear-cut answer but, in general, the decision will be subjective as it is a matter of taste how much freedom is left to the designer.

Experience shows that under-and over-specification are two of the main classes of problems that arise in an RS and it is recommended that these are investigated.

Underspecification is often discovered only when the designer's interpretation is implemented in a way that is unacceptable or unintended. Production of a model or animation of the specification is the best way to test for underspecification.

Overspecification will be spotted more easily by examining the reason for each requirement. It should be noted that it is possible for the RS to be underspecified in some areas and over specified in others.

Completeness and validity can best be appraised via a review with the purchaser. Some guidance on conducting reviews is given in Appendix B. The review may be supported by demonstration of a prototype showing the primary facets of the system specified.

8 Use of Methods and Tools

The STARTS message applies equally well to Requirements Specification as it does to other stages in the life cycle:

> "The application of appropriate methods and tools can reduce the risk in producing a Requirements Specification and improve its quality."

8.1 Informal Approaches

Informal "methods" or approaches employ an RS notation which has no fixed syntax or semantics. Typically, the notation used is English supported by ad hoc diagrams with no pre-defined meaning. It is rare for an informal approach to have any defined procedures for requirements elicitation, verification or validation.

The advantage of an informal approach is that it is easy to apply and requires little, if any, training. An apparent advantage is that the RS is easy to understand but this is illusory and informal specifications are usually ambiguous, inconsistent and incomplete. Many RSs are produced using informal approaches; this is the source of many problems in system developments.

8.2 Semi-Formal or Structured Approach

A semi-formal or structured approach typically uses diagrams with well-defined syntax and semantics. Most structured methods also have a well-defined set of procedures and guidelines for carrying out the major activities in RD.

Structured approaches, for example CORE, SADT, offer much greater scope for achieving consistency, completeness and clarity in an RS; they are usually good vehicles for communication between user and analyst. Their main drawback is the limited expressive power of their notation so that there are certain facets of requirements which it is difficult to articulate precisely using these approaches. People also require training to read or write the selected notation.

8.3 Formal Methods

Formal methods, for example, VDM, Z, use a precise mathematical notation which can be highly expressive. However, experience has shown that it is difficult to train users and purchasers to read or write formal specifications unless they are mathematically inclined. There are ways of overcoming this problem, for example, via animation, but formal methods are a relatively new technology and do not yet enjoy widespread application. Even so, there is a growing interest in them, particularly in the area of safety critical and high integrity software. This is discussed in Appendix C).

In addition, formal "methods" are often little more than notations and it is rare for them to include comprehensive guidelines for elicitation and validation. Thus, in the methodological sense, they are often weaker than structured methods.

In the long run, it seems likely that formal methods will become more widely used for requirements specification. However, before this happens they will probably evolve substantially, for example, to have much stronger links with structured methods. This evolution will require research into a number of aspects, for example, the ability of formal methods to specify safety properties and validation.

This chapter ends with a set of checklists which can be used as a quick reference guide.

Extended Checklists for the Production of an RS

In addition to the structure/content of the RS, these extended checklists cover the preliminary RS activities and give three levels of guidance for use during the subsequent RS process. They summarise the important issues discussed in Chapter 10 and Appendix B and should be used as a guide to ensure that the stages of Requirement Specification are adequately addressed.

Where more detail can be obtained from this chapter or Appendix B, a reference is included. These are to section numbers, (prefixed with "10" for this chapter) but not all points are cross-referenced since some are the concern of other chapters and some are a distillation of a number of issues covered in this chapter or Appendix B.

Structure of the Extended Checklists

A - PRELIMINARY ACTIVITIES
B - THE REQUIREMENT SPECIFICATION PROCESS

1 General Guidelines for RD
2 Structure and Content of the RS
3 Guidelines for Writing Requirements

Checklist Descriptions

CHECKLIST A	Summarises the important issues, discussed in chapter 10 and Appendix B, which should be addressed prior to starting production of the RS.
CHECKLIST B.1	Summarises the important issues, discussed in chapter 10 and Appendix B, concerning RD.
CHECKLIST B.2	Outlines the structure and content of a two level RS.
CHECKLIST B.3	Provides guidelines to writing the body of the RS within the structure outlined in checklist B.2.

Checklist A

TOPICS to ADDRESS BEFORE STARTING the RS PROCESS

1. Has the purpose of the system been clearly established? *10.3.1 and B.5.1 section 1.1*
2. Has the procurement strategy been clearly defined? *10.3 and Chapters 4 and 9*
3. Have configuration/change control procedures been defined? *10.3.3*
4. Has the use of method(s) for requirements capture and analysis and their associated training requirements been addressed? *10.5.3*
5. Are there any safety critical features of the proposed system and, if so has the use of formal methods been considered? *Chapter* 7
6. Who has been given the task of producing Levels I and II? *10.5.3*
7. Have review procedures, constitutions of review bodies and review schedules been defined? *Chapter 4*
8. Has a model text (i.e. outline document structure) for the RS been agreed?
9. Have all the relevant viewpoints of the system being specified been identified? *10.5.3*
   For example:
   * Who will use the system?
   * Who will operate the system?
   * What will interface with the system?
   * Who will maintain the system?
   * In what physical environment will the system be expected to operate?
10. Has a list of agreed terminology been produced? *10.5.3*
11. Has the proposed system been assessed for feasibility? *10.5.3*
12. Has a Quality Plan been defined? *Chapter 5*
13. Who is the approval authority? *10.5.2*

Checklist B

Checklist B.1 - General Guidelines for RD

This checklist consists of a list of points to remember.

CHECKLIST B.l

1. The RS should be produced bearing in mind that it is the sole means of communication between the purchaser and the supplier. *10.1.1*
2. The RS must be written from the purchaser's viewpoint, that is, it should state WHAT the system is required to do NOT how it should do it. *10.4.2*
3. The RS fulfils two roles: *10.1.1*
   * the primary input to the design process
   * the baseline for acceptance testing.
4. Level I of the RS should concentrate on defining the requirement from the viewpoint of its operating environment, that is, defining the essential system functions (cardinal points) and their interface with the outside world. *10.6.1 and B.5.1*

Note: Level I should NOT address the relationships between the functions that comprise the target system.

5. Level II of the RS should concentrate on defining the system requirement with respect to internal viewpoints of the system, for example, sub-systems, dependabilities, security, safety. *10.6.2, B.5.3 and B.5.4*

Checklist B.2 - Structure and Content of the RS

For the existing Level II structure and content see Appendix B.5.2

Checklist B.3 - Guidelines for Writing Requirements

In addition to the checklist below, see sections B.4 and B.5 in Appendix B.

GUIDELINES for WRITING the RS

1. Uniquely reference each statement. *10.6.2*
2. Keep individual requirement statements short, for example, less than 250 words. *10.7.1*
3. Express requirements; avoid offering design solutions. *10.4.2*
4. There should be no duplicate requirement statements, for example, avoid restating a requirement in order to explain a new one. *10.7.1*
5. Requirements should be stated in a testable form, for example, avoid a statement such as "the system should be user friendly". *10.7.1*
6. If design constraints are specified, clearly define which functions are affected. *B.5.1 section 1.3*
7. Maintain a consistent level of detail across the RS. *B.4.2.8*
8. Ensure each requirement is traceable to its parent requirements, for example, requirements expressed at Level II should be derived from Level *1.10.6.1*
9. State only feasible requirements. *10.5.3*
10. Ensure each requirement reflects the capabilities of system design technology. *10.5.3*