Chapter 2 : The Role of Requirements Engineering Techniques

Objectives

- To discuss Requirements Engineering (RE) in relation to known causes of system failure
- To discuss the role of RE techniques in supporting the RE process
- To discuss the role of RE techniques in supporting the development of knowledge throughout the process
- To discuss the role of RE techniques in supporting human communication
- To highlight the problems associated with managing the RE process
- To provide a 'wish list' of requirements for RE techniques which can be referred to throughout the book
- To provide a rationale for the choice of techniques in chapters 3, 4 and 5

2.1. Introduction

According to Pohl's definition of Requirements Engineering (RE) (Pohl, 1993)

"Requirements Engineering can be defined as the systematic process of developing requirements through an iterative co-operative process of analysing the problem, documenting the resulting observations in a variety of representation formats, and checking the accuracy of the understanding gained."

This provides a good definition of the RE process, and a starting point for considering what RE techniques[1] might be needed. Kawalek, 1993, defines a process as:

"a set of identifiable, repeatable actions which are in some way ordered and contribute to the fulfilment of an objective".

In their paper describing a 'Process Engineering Framework', Kawalek and Wastell, 1994, ask what we are to expect from a 'process' and state that they are interested in processes because "businesses need to be able to design and maintain the way they work in order that they can operate effectively and flexibly".

In Requirements Engineering we are also interested in process and many recent research papers are concerned with identifying suitable processes (Bell and Oates, 1994, Dobbin and Bustard, 1994). While others are concerned with process improvement and with measuring the success of RE processes (El Emam and Madhavji, 1995, Lubars et al.

[1] the word 'techniques' means a method of performing or executing some specialised activity (Penguin English Dictionary) and is used throughout this book as a generic term for methods, tools, models, etc.

1993). Much of the work is concerned with identifying the “repeatable actions which are in some way ordered” (Kawalek, 1993).

The second part of Kawalek’s definition refers to process as contributing to “the fulfilment of the objective”. What is the objective for the RE process? It could be argued that the objective is to produce a specification of requirements. This specification being some document which describes what needs to be designed. It could equally well be argued that the objective of the RE process is to specify a system which will ultimately be successful.

Bubenko, 1995, recognises that most of the problems in system development have their roots not just in the technical (software) issues but also in managerial, organisational, economical and social issues.

In his paper Robinson, 1994, states that as many as 50% of information systems projects may be considered as failures. However, following his study of the failure of the systems in the London Ambulance Service, Robinson concludes that:

“ Firstly the failure or success of a project will always be defined in relation to a particular group with its own interests, roles, goals and expectations........
Secondly, these interests and goals are defined in the total context of an organisation and its political and social environment and not just in relation to the technology.”

Thus indicating that the notion of success is not straightforward but that it can only be defined in relation to particular groups and within the context of a particular organisational setting.

In 1987, Lyytinen and Hirschheim undertook a survey of empirical literature on information systems failures and concluded that they could be classified into four types:

1. Correspondence failure: correspondence failure means that design objectives have not been met.
2. Process failure: process failure relates to the information systems development process where budgetary, time or other resource allocations have overrun to the point where any benefits expected from the proposed system have been negated, or where the allocated resources do not result in a workable system.
3. Interaction failure: interaction failure is the argument that the low level use of the system can be interpreted as failure.
4. Expectation failure: expectation failure is simply that the system has failed to meet the expectations of at least one stakeholder group.

Thus if the objective of the RE process is to specify a successful system then the requirements engineer needs to be aware of the possible causes of failure and must use techniques which will help avoid failure. In this chapter an attempt is made to identify what those techniques should be. The discussion which follows is based on Lyytinen and Hirschheim's classification by linking possible causes with types of failure.

Correspondence failure is not discussed, because it is argued that the purpose of the requirements engineering process is to set the design objectives. Thus this type of failure is not considered.

Table 1 below shows process failure linked to three possible causes: lack of a systematic RE process, poor communication between people and poor management of people and resources. Interaction failure and Expectation failure also link to three possible causes: poor communication between people, lack of appropriate knowledge or shared understanding and inappropriate, incomplete or inaccurate documentation.

Possible Cause / Type of failure	lack of a systematic process	poor communication between people	lack of appropriate knowledge or shared understanding	inappropriate, incomplete or inaccurate documentation	poor management of people or resources
Process	¦	¦			¦
Interaction		¦	¦	¦	
Expectation		¦	¦	¦	
Section headings	2.2 Process	2.3 Human Commu-nication	2.4 Knowledge Develop-ment	2.5 Docum -entation	2.6 Manage-ment

Process failure is discussed under three section headings, firstly the requirements engineering process itself, secondly management of the process, and human communication.

Interaction failure and expectation failure are discussed under three section headings. The first is 'human communication within requirements', here it is argued that many of the problems relating to poor system use and failed expectations are due to the fact that the people involved in the RE process do not communicate sufficiently or effectively with each other. The second heading is ‘knowledge development’, the problems relating to poor system use are due to a lack of knowledge about what the system should do, and in particular, the inadequacy of the type of knowledge acquired about the users present job, the technological options and the future situation. In addition to knowledge, people must also develop an understanding of what is needed. This knowledge and understanding will form the basis for stakeholder expectations of the system. The third heading is ‘documentation of requirements’, this is included because documentation is one representation of what the proposed system should do and, as such, is a description of stakeholder expectations of the system.

Thus the remainder of this chapter is an attempt to discuss some of the problems associated with requirements engineering and from these to derive an number of requirements for RE techniques.

The problems are discussed under five headings:

1. *The requirements engineering process*. (a cause of process failure)
   The RE process is described as a series of activities which result in the development of intermediate workproducts.

2. *Human communication within requirements*. (a cause of interaction and expectation failure)
   A key factor in specifying a successful system is understanding the needs of users and other stakeholders. This section discusses the problem of communication between users and requirements engineers, and the problem of identifying the needs of different groups of stakeholders.

3. *Knowledge development* (a cause of interaction and expectation failure)
   Another key factor in specifying a successful system is that all the stakeholders (including designers and requirements engineers) should develop appropriate knowledge and understanding. This section is concerned with the areas of knowledge that need to be developed as part of the RE process. It is concerned with abstract representations and with human understanding.

4. *Documentation of requirements* (a cause of interaction and expectation failure)
   The objective of the RE process may be a specification of requirements (for the design), but there may also be other requirements documents produced. This section discusses issues related to market requirements and to user requirements.

5. *Management of 1,2,3 & 4* (a cause of process failure)
   The success of the RE process itself will also depend upon it being well managed. This section discusses some of the issues related to management.

Figure 1 shows the structure of this chapter.

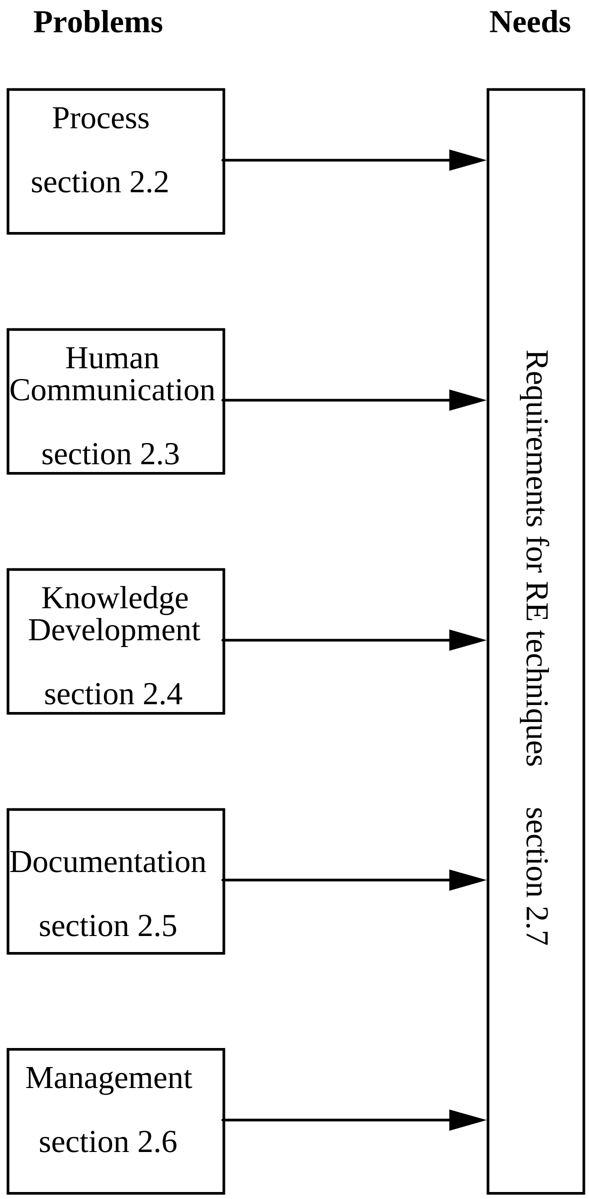


fig 1 structure of this chapter

The chapter concludes in section 2.7 with a list of seventy requirements for RE techniques. These are referred to throughout the text.

2.2. The Requirements Engineering process

As described in chapter one, in general terms, the RE process can be thought of as a series of activities consisting of articulating the initial concept, problem analysis, feasibility and choice of options, analysis and modelling and requirements documentation. Each activity will require the use of potentially different techniques.

Often the system development process is described in terms of work products, for example, SSADM (Downs et. al., 1992) is described in terms of numbers of stages, each stage has a number of steps, each step has associated inputs, processes and outputs. In line with many methods the emphasis is placed on process and outputs or workproducts. This approach is attractive in situations where the development process is being audited since workproducts can be seen and scrutinised as part of a quality assurance process.

There are a number of arguments in support of a systematic approach to requirements engineering. One is the ability to control the project by producing standardised maintainable outputs (Bjorn-Anderson, 1984, Glasson, 1984, Rzevski, 1980).

A second argument is that once a systematic procedure is in place it should be possible to measure the effectiveness of that procedure and hence to seek ways of improving it (Dale, 1982, Skousen, 1982, Wasserman, 1983).

A third argument is that it lends itself to the use of automated aids. Automated tools are considered important because it is assumed that they increase productivity and reduce administrative costs. (Bantleman, 1985, Brandt, 1983, Brodie et. al., 1983)

Thus the RE process, in general terms, consists of a series of activities. Each activity may result in a workproduct. The workproduct should be capable of being maintained and be subject to quality control. The effectiveness of the RE process should be capable of measurement and improvements in the process should be quantifiable. The use of automated tools is desirable.

Thus RE techniques are needed which:

- support articulation of the product concept
- support problem analysis
- support feasibility studies and cost-benefit analyses of options
- support analysis and modelling
- support documentation of requirements
- support a systematic step by step approach
- provide standardised ways of describing workproducts
- provide procedures for maintaining workproducts
- provide ways of assessing the quality of workproducts
- enable identification of measures and measurement of the RE process
- supports descriptions of effectiveness in RE terms
- support analysis of opportunities for process improvement
- provide automated support for the RE process

This discussion has focused on the RE process as a ‘set of identifiable, repeatable actions’. The next section deals with issues of who should be involved in the RE process and how they communicate with each other.

2.3. Human Communication within Requirements

The discussion below is based on Macaulay, 1993, in which a variety of issues associated with human communication are described. It is structured under the headings (I) users are consulted; (ii) users participate; (iii) stakeholders participate and (iv) stakeholders communicate.

2.3.1 Users are Consulted

Firstly the most traditional approach is to think of the requirements engineer as responsible for 'eliciting' requirements from users. This is usually achieved through use of interviewing, questionnaires or by observation, where the user plays a relatively passive role. In structured analysis approaches, such as SSADM (Downs et al., 1992), user views are elicited at appropriate points in the method. The method relies on the expertise of the requirements engineer to model present activities, to elicit requirements and to develop a vision of the future to present to the project manager and other stakeholders.

Other approaches explicitly seek to identify the viewpoints which must be incorporated, for example CORE (Mullery, 1987), requires the requirements engineer to identify the 'customer authority' and the 'viewpoint authority'. Each 'viewpoint authority' is the person responsible for providing the analyst with the information needed for some particular 'view' of the problem domain. That 'view' could be that of the end-user or of the user manager or , as is often the case, the 'view' of plant, machines or controllers or some already existing computer system. The 'viewpoint' must be responsible for processing information, it must receive input from some other viewpoint and send output to a viewpoint. Thus although a user viewpoint might be a valid viewpoint there is a tendency for users needs to be considered only in as much as they have needs as information processors.

In contrast to the data and process oriented approaches the object oriented approach is now increasing in popularity. In particular in Object Oriented Analysis (Coad & Yourdon, 1991) it is suggested that this approach improves analyst and problem domain expert interaction because object oriented is a natural way of thinking. The five stage method recommended by Coad and Yourdon still prescribes a passive role for the user, with the traditional view of users as sources of information and reviewers of models developed.

The methods discussed above assume the requirements engineer is responsible for understanding the problem domain and that users are a source of information and not normally active participants in deciding the requirements of the proposed system.

The techniques employed within the above methods ensure that users are consulted but do not encourage users to actively participate in the decision making process.

2.3.2 Users Participate

According to Avison & Wood-Harper 1991, in participative approaches all users are expected to contribute to and gain from any information system, and that participation should increase the likelihood of success. Participation can take many forms, for example, in ETHICS (Mumford, 1986, 1989), the users assist in analysing their problems at work, complete job satisfaction questionnaires and set future objectives for efficiency, effectiveness and job satisfaction. Eason (1988, 1989) on the other hand defines three categories of users whose needs should be taken into account. Primary users, who are those likely to be frequent hands-on users of the proposed system, secondary users, who are occasional users or those who use the system through an intermediary, and tertiary users, who are those affected by the introduction of the system or who will influence its purchase but who are unlikely to be hands-on users.

In an attempt to smooth the transition from requirements to design the formation of a 'design team' is recommended. More specifically Eason (1988) offers a number of options for the construction of the design team (who also have responsibility for requirements) where the roles of the 'technical experts' and the 'customers' are clearly identified. The technical experts contribute their skills to the creation of a system whilst the customers are concerned with the world they will have to inhabit after the change caused by the new system. The customers also have a wide range of specific knowledge about the way the organisation functions and the tasks it undertakes. The technical experts will want the system to help them advance their own design skills. Eason recommends therefore that the structure of the design team recognises the fact that both specialists and customers have expertise to contribute and vested interests in the solutions adopted.

Evaluation Criteria	Options		
	a	b	c
1. Specialist technical skills where needed	¦	¦	X
2. Specialist social skills where needed	X	¦	¦
3. Users able to contribute task knowledge	X	¦	¦
4. Users able to assess organisational effects	X	¦	¦
5. Stakeholders able to negotiate interests	X	X	¦
6. All users develop feelings of ownership	X	X	¦
7. Practical use of resources	¦	¦	X
8. Acceptable to the organisation	¦	¦	X

Figure 2 Alternative team structures, after Eason, 1988.

The three options suggested by Eason are:

a) *Technical Centred Design* where customers commission and accept the system and are informed and consulted throughout the design process;
b) *Joint Customer-Specialist Design* where user representatives are involved in all stages of the design process; and
c) *User-Centred Design* where the technical experts provide a technical service to the users and all users contribute to the design.

In his discussion on the alternative design team structures Eason suggests a number of criteria that could be used to evaluate the effectiveness of each structure, presented in figure 2. The first two criteria are concerned with the presence of technical skills needed and with the human and organisation specific knowledge needed if the proposed system is concerned with organisational change. Three and four refer to the expert contributions that can be made by potential users, particularly the extent to which users have the opportunity to contribute specific task knowledge or to assess the organisational effects of the proposed system. Five and six are concerned with the vested interests of the different stakeholders, for example, are stakeholders able to negotiate their interests and are users able to develop a feeling of ownership? The last two criteria deal with the practicality and acceptability of the design team structure as far as the commissioning organisation is concerned.

Eason's own evaluation of the alternative team structures suggests that each approach has strengths and weaknesses. The Technical Centred Design Team scores favourably on having the technical skills where needed, is a practical use of resources and acceptable to the commissioning organisation but fails on every other criteria. The Joint Customer Specialist Design Team on the other hand passes on all criteria except five and six whereby stakeholders can't negotiate interests and users cannot develop a feeling of ownership. The third option, the User Centred Design Team, scores favourably on most counts. It most noticeably fails, however, on the last two criteria: it is not perceived as a practical use of resources and is generally not acceptable to the commissioning organisation.

None of the structures proposed is ideal. The Technical Centred design team finds favour with the commissioning organisation but largely ignores the need for participation of users and other stakeholders. The Joint Customer-Specialist design team is widely accepted but is likely to result in some stakeholder needs being ignored. The User-Centred design team, on the other hand, takes everyone needs into account but is too inefficient in the use of resources.

User participation is widely recommended by those concerned with socio-technical design and as can be seen from the above discussion the ability of users to participate effectively is determined as much by the structure and remit of the design team as by the provision of suitable techniques for allowing their views to be incorporated.

2.3.3 The Stakeholders Participate

A stakeholder is defined here using Mitroff's (1980) terms, as all those who have a stake in the change being considered, those who stand to gain from it, and those who stand to lose.

The stakeholders in any computer system fall into four distinct categories (Macaulay, 1993) :

1. Those who are responsible for its design and development, for example, the project manager, software designers, communications experts, technical authors;

2. Those with a financial interest, responsible for its sale or for its purchase, for example, the business analyst, the marketing manager, the buyer;

3. Those responsible for its introduction and maintenance within an organisation, for example, training and user support staff, installation and maintenance engineers and user managers;

4. Those who have an interest in its use, for example, user managers and all classes of users, that is, primary, secondary or tertiary .

Some of the stakeholders identified above, particularly in categories (a) and (c) have a direct responsibility for the design and development of the various system components and hence have a major interest in being involved in the requirements process. Those in category (b) have a financial responsibility for the success of the computer system and therefore may also need to be involved. The stakeholders in category (d) will be the recipients of the resulting computer system, they also have a major contribution to make in terms of specific task knowledge and the ability to assess the likely effects of the new system.

Despite Eason's observations concerning the acceptability of stakeholder participation in requirements and design there is an increasing recognition of the need to develop a shared meaning of the system being specified and designed. For example, Konda et al. (1992) argue that increasing design effectiveness is essentially increasing the breadth and depth of a shared meaning between the designers participating in the process of a specific design situation.

Reich et al. (1992) extend this notion to include not only different experts in creating this shared meaning but also a range of non-expert designers such as users, resellers, maintainers. They also argue that extracting needs from users is a dynamic ongoing activity where the central purpose is to continually evolve the design on the basis of ‘multilateral participation of all relevant actors’.

The problem of multilateral participation of a range of non-expert designers is one of potentially incompatible perspectives and conflicting objectives. Thus it is argued here that while stakeholder participation is desirable it will not necessarily lead to agreement or consensus as to the way forward.

2.3.4 The Stakeholders Cooperate

In this case not only do stakeholders participate in requirements but they also cooperate with each other and are actively involved in making decisions as to the scope of the proposed new system. Similar arguments for cooperative design have been put forward by Pasch, 1991 in a paper on ‘Dialogical Software Design’ in which he states that :

“Software development is not merely a mathematical or technological challenge, but a complex social process, in which the kind of communication and cooperative, creative interaction of the participants determine the quality of the collaboratively developed product.”

He proceeds to argue that design requires experience, intuition, imagination, and common sense and that the design process is guided by insights from all participants. If this is true in software design then surely the case for cooperative in the requirements process can be argued even more strongly.

Stakeholder	*Motivation*	*Expertise*
Software Designer(1)	to produce a technically excellent system, & use latest techniques	in latest techniques & creative design skills
Software Designer(2)	to reuse existing software tools or designs	knowledge of existing systems
Systems Analyst	to produce requirements specification on time	expertise in problem analysis
Technical Author	to develop learning materials which meet user needs	authoring skills, documentation design
User Representative	to introduce change with minimum disruption & maximum benefit	knowledge of organisation, users and tasks
Training & User Support Staff	to support existing accounts & in order to generate future revenue	knowledge of current user problems
Business/Market Analyst	to be 'better' than the competition	knowledge of business/ market needs
Project Manager	to successfully complete the project within given resources	knowledge of project planning & of previous overrun projects

Figure 3 Expertise and motivation of stakeholders in a system development project

Whilst it can be argued that the requirements process would be enriched by cooperation between stakeholders representative of the four categories described above, it is by no means clear that interaction between people with such a diversity of expertise and motivations, see figure 3, would result in anything but chaos. For example Westley and Walters (1988) identify five Generic Meeting Problem Syndromes which could arise were such a requirements capture ‘team’ to meet to discuss requirements for a new system . These are the 'multi-headed beast', the 'feuding factions', the 'dominant species', the 'recycling meeting' and the 'sleeping meeting'. The 'multi-headed beast syndrome', for example, can arise when there is no agreement on the agenda or when group members attempt to mix problem solving strategies because there is no listening taking place and no integration of ideas. Another syndrome identified is that of 'feuding factions' where arguments are repeated, subgroups form within the meeting and there are hidden agendas being pursued. The 'dominant species' syndrome is often witnessed at design meetings where one member of the group attempts to dominate the rest, other members can become withdrawn, afraid or frustrated.

In order for stakeholders to cooperate therefore, it is argued that meetings need to be facilitated in some way in order that they might agree on the agenda, agree on which problem solving strategies to adopt at given points in the discussion and that all stakeholders are given the opportunity to participate.

In addition to facilitated meetings the stakeholders need techniques which will encourage multiparty interaction and provide a focus for discussion and decision making. Examples of such techniques can be found in the QFD method (Quality Function Deployment) where a large matrix called the 'House of Quality' is drawn up by the requirements team in order to map the customer requirements on to the proposed product characteristics and features (Sullivan, 1986). The strength of the relationship between what the customer wants and what the supplier is intending to provide is entered in each cell of the matrix. Further analyses can be undertaken using the matrix as the focus for prioritising requirements and competitor analysis. Reports on the usage of QFD claim that it encourages interaction and helps build consensus and shared team understanding (Burrows, 1991). Further examples of similar techniques which encourage human communication can be found in the HUFIT toolset (Taylor, 1990), JAD (Joint Application Design) workshops (August, 1991) and Cooperative Requirements Capture workshops (Macaulay, 1994).

Thus the human communication problem concerns not only selection of personnel but also the means by which the people communicate with each other.

Thus techniques are needed which:

- provide guidance on interviewing users
- provide guidance on the design and use of questionnaires
- provide guidance on conducting observations of users
- support identification of various viewpoints
- support reconciliation of viewpoints
- support the user in reviewing models developed
- support users in analysing their own problems and identifying the need for change
- support construction of appropriate requirements teams
- support identification of stakeholders
- support the development of a ‘shared meaning’ of the system being specified
- encourage intuition, imagination and common sense among participants
- support communication between people from a diversity of backgrounds
- support facilitated meetings with predefined agendas and problem solving strategies
- support the development of listening skills among participants

Communication between people is important throughout the RE process in order to produce the documents and other workproducts required. Another important aspect of the RE process is the development of knowledge and understanding among participants. The next section discusses the areas of knowledge that need to be developed.

2.4. Knowledge Development

The results of the development process are, according to Kensing and Munk-Madsen (1993), “a system and a complete technical and organisational implementation process”. They argue that the intermediate results are not only documents but also knowledge obtained by the participants and that regardless of the development model, be it waterfall, spiral, incremental or parallel, these intermediate results form the basis of important decisions.

Knowledge is developed by the people who are involved and different types of knowledge are needed. Within the requirements process a ‘vision’ of the future system needs to be acquired, knowledge of users current practices is needed, projections of change using knowledge of the organisation and of external factors, knowledge of skills and motivations of the targets users is needed, indeed there are many areas of knowledge required.

Kensing and Munk-Madsen, (1993), suggest that six areas of knowledge and understanding are needed before system development begins. These areas are based on the thesis that:

“ The main domains of discourse in design are:

- * users’ present work
- * technological options
- * new system

Knowledge of these domains must be developed and integrated in order for the design process to be a success.”

In addition they suggest that “two levels of knowledge” of each of these domains of discourse is required. These are:

1. Abstract Knowledge: to get an overview of the domain of discourse
2. Concrete Experience: in order to understand that abstract knowledge

Thus leading to the six areas of knowledge shown in figure 4.

Areas of Knowledge	*Abstract Knowledge*	*Concrete Experience*
Users' Present Work	1 Relevant Structures on users' present work Users & Developers need	4 Concrete experience with users' present work Users have, Developers need
New System	2 Visions and Design proposals Users & Developers need	5 Concrete experience with new system Users need
Techno-logical Options	3 Overview of Technological options Developers need	6 Concrete experience with technological options Developers have, users need

Figure 4, Six areas of knowledge, adapted from Kensing and Munk-Madsen, 1993

Each of these areas of knowledge need to be developed as part of the requirements engineering process. At the beginning of the process some knowledge is already possessed, for example, the users have concrete experience of their present work (area 4) and the developers have concrete experience of technological options (area 6). Techniques are needed to facilitate sharing of this knowledge, users with developers as in area 4 or developers with users as in area 6. Other areas of knowledge such as area 1 & 2 both users and developers need to acquire a common vision of the future system and agree on relevant abstract structures of the users present work.

The type of technique used will influence the area of knowledge which can be developed and the nature of the communication between users and developers. Figure 5 shows a range of techniques and classifies them according to the six areas of knowledge described in figure 4.

Tools and Techniques for Knowledge Development	*Abstract*			*Concrete*		
	1	2	3	4	5	6
Observations				¦		
Interviewing Users	¦			¦		
Developers doing users' work				¦		
Videorecording				¦		
Mock-ups				¦	¦	
Think-aloud experiments				¦	¦	
Drawing rich pictures	¦			¦		
Ethnographic studies	¦			¦		
Object-oriented analysis	¦	¦				
Event lists	¦	¦				
Entity-relationship diagrams	¦	¦				
Future workshops	¦	¦				
Conceptual modelling	¦					
Dataflow diagrams	¦	¦				
Card Games				¦	¦	
Formal Language Specifications	¦					
Prototyping		¦			¦	¦
Visits to other installations			¦			¦
Literature Study			¦			
Study of standard software			¦			¦

Figure 5, Tools and Techniques for Knowledge Development
(adapted from Kensing and Munk-Madsen, 1993)

The choice of techniques within any given development project will affect the capability of the requirements team (users, developers and other stakeholders) to develop a shared understanding of the users' present work, the technological options and the future system. Ideally a range of techniques should be employed on any given project so that all six areas of knowledge are developed.

It is interesting to note that many 'traditional' requirements techniques tend to favour the development of abstract knowledge. Techniques for developing concrete experience are less well represented.

The work of Kensing and Munk-Madsen provides an excellent framework for identifying the role of requirements techniques. However, the proposed system often has strategic implications for a company and thus additional areas of knowledge may need be developed.

Figure 6 illustrates the fact that when embarking on the requirements task it is necessary not only to develop visions of the future system but also to develop visions of the future state of markets, of competitors and competitor products, of company strategy, of planned or projected organisational change, and of course changes in government policies and legislation.

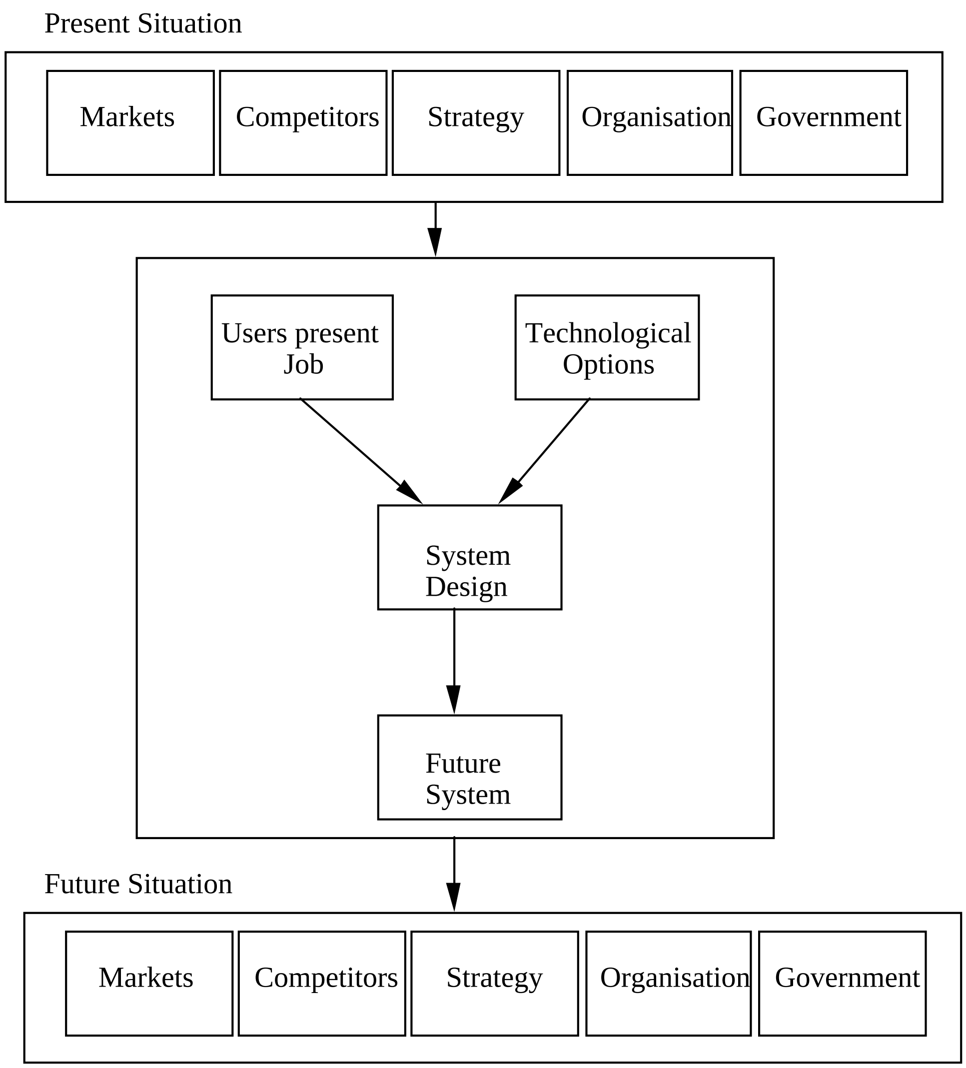


Figure 6 Areas of knowledge related to requirements engineering

Thus there is a need for requirements techniques which support the development of:

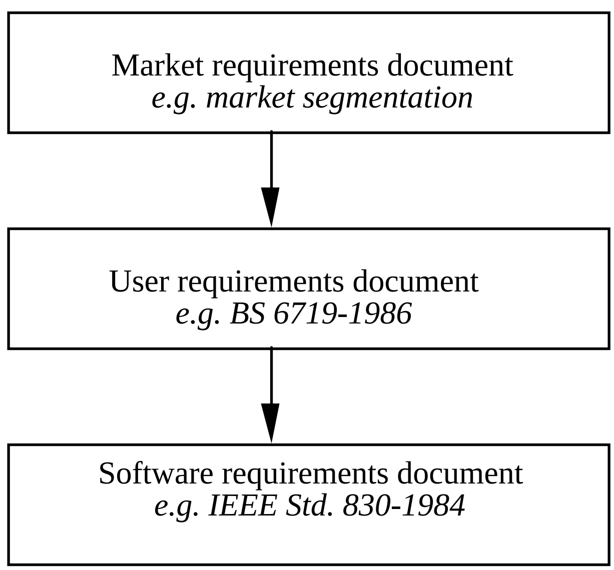
- Relevant structures on the users' present work
- Visions and design proposals

- Overviews of technological options
- Concrete experience with the users' present work
- Concrete experience with the new system
- Concrete experience with technological options
- Knowledge of the current market and insights into future market changes
- Knowledge of current and proposed competitor products
- Knowledge of company strategy and likely future developments
- Knowledge of the current organisation and the potential for change
- Knowledge of government policy and planned changes

How people communicate and develop knowledge are important to the success of the RE process. The next section discusses the role of the requirements document in the RE process.

2.5. The Requirements Document

Most companies develop their own standard form and content of the requirements document to meet their own needs and purposes. There may be different types of requirements document, figure 7 for example, shows three types, each of which contains different information.



Three different types of requirements document
Figure 7

A market requirements document would be used in situations where a supplier is wanting to develop a (generic) product which can be used by potentially a large number of customers, for example, a point of sale terminal for use in supermarkets and chain stores.

A typical market segmentation statement will contain the following:

1. Industry type : a description of the industrial groupings or types of company targeted by the proposed product.

2. Size : a description of the size of the market in terms of the number of buying organisations, number of sites, employees, number of potential product users.

3. Geography : a description of any geographical differences which will affect the proposed product, such as, language differences, accounting systems, cultural differences, organisational structures, government regulations, standards.

4. Frequency : a description of the projected frequency of purchase and frequency of use of the proposed product, including usage patterns and purchase of upgrades.

5. End use : a description of the likely end use of the proposed product, including a description of what users do now and how it is likely to change. A ‘Day In the Life Of’ a typical user or group of users is often used.

6. Product specification : a description of the proposed product in terms of its objectives, its use and its impact on the customers business.

7. Buyer’s and user’s identities : a description of the people likely to be responsible for purchase of the proposed product, including those likely to recommend its purchase within the customer organisations.

8. Source loyalty : a description of customer loyalty to existing work practices, to existing suppliers or other factors which may affect the acceptance of the proposed product into the organisation.

9. Buyer’s and user’s personality : information which will assist in the marketing of the proposed product, including style of promotional messages and sales channel.

In contrast to this a user requirements document would be used by a customer who wants to describe their requirements for a system. It may be that the system will be an off-the-shelf package, a package which will need to be modified to meet the requirements, enhancements to existing systems or a bespoke system.

An example of a typical user requirements document is the BSI British Standard Guide to specifying user requirements for a computer based system (BSI, 1986). The standard is in two parts. The first part is a generalised summary of requirements which will introduce potential suppliers to the customers organisation, the nature of the problem, and the principle constraints such as cost, time and security. The second part specifies all the functions the system is required to perform, when and in what sequence. It also includes details of acceptance criteria and performance monitoring. The BSI guide also offers advice on how to evaluate supplier responses.

Other examples of recommended contents lists for a user requirements document exist in the HCI (Human Computer Interaction) literature. For example, Macaulay, 1995b suggests:

1. Management Summary (including the business case and a brief description of the proposed system)

2. The Human Requirements (Description of the objectives of the commissioning organisation, List of the stakeholders together with their objectives, List of key workgroups and users and their objectives)

3. The High Level Functional Requirements (List of work roles to be supported and why, Description of each work role in terms of users, objects and tasks)

4. The Detailed Functional Requirements (Consolidated list of objects to be supported, Descriptions of each object together with details of user tasks associated with each object)

5. The Quality Attributes (usability, reliability, portability, performance, security, maintainability, acceptability or others depending on the proposed system)

6. Organisation and User Assistance Requirements (User documentation requirements, Training requirements, User support, Human computer interface requirements)

7. The Technological Requirements and Constraints (Known hardware requirements (user or supplier), Known software constraints (user or supplier))

An example of a software requirements document is the IEEE Std 830-1984 taken from the IEEE Guide to Software Requirements Specifications (IEEE, 1984). A typical table of contents will include:

IEEE Table of Contents:

1. Introduction
    - 1.1 Purpose
    - 1.2 Scope
    - 1.3 Definitions, Acronyms, and Abbreviations
    - 1.4 References
    - 1.5 Overview
2. General Description

2.1 Product Perspective
2.2 Product Functions
2.3 User Characteristics
2.4 General Constraints
2.5 Assumptions and Dependencies
3. Specific Requirements
3.1 Functional Requirements
3.1.1 Functional Requirement 1
3.1.1.1 Introduction
3.1.1.2 Inputs
3.1.1.3 Processing
3.1.1.4 Outputs
3.1.2 Functional Requirement 2
...
3.1.*n* Functional Requirement *n*
3.2 External Interface Requirements
3.2.1 User Interfaces
3.2.2 Hardware Interfaces
3.2.3 Software Interfaces
3.2.4 Communication Interfaces
3.3 Performance Requirements
3.4 Design Constraints
3.4.1 Standards Compliance
3.4.2 Hardware Limititations
...
3.5 Attributes
3.5.1 Security
3.5.2 Maintainability
...
3.6 Other Requirements
3.6.1 Database
3.6.2 Operations
3.6.3 Site Adaptation
...
Appendices
Index

A Requirements Document is a specification of what a computer system is required to do, (not how it will do it). A Requirements document can be evaluated by it's effectiveness as a means of communication, by it's contents as measured against a checklist and by the quality of the statements it contains.

According to the IEEE Std 830-1984 a "good" Requirements Document should contain statements which are unambiguous, complete, verifiable, consistent, modifiable, traceable and usable during the operation and maintenance phases.

Unambiguous
Requirements are often written in a natural language where statements can have more than one meaning. Formal requirements languages help reduce ambiguity, because the formal language processors automatically detect many lexical, syntactic, and semantic errors.

Complete
The Requirements Document is complete if it includes all of the significant requirements, whether relating to functionality, performance, design constraints, attributes or external interfaces and conforms to the company standard.

Verifiable
An example of a non verifiable requirement:
"The product should have a good human interface"
An example of a verifiable requirement:
"The system will respond to a user request within 20secs of user pressing the 'enter' key, 80% of the time"

Consistent
Three types of conflict which can occur:

- different terms used for the same object, for example ‘a P45’ and ‘a tax form’ might be used to describe the same form.
- characteristics of objects conflict, for example, in one part of the requirements document : 'a red light will indicate a fault', while in another part 'a blue light will indicate a fault'.
- logical or temporal faults, for example: 'A follows B' in one part, 'A and B occur simultaneously' in another.

Modifiable
The requirements document should have a coherent and easy-to-use organisation, with a table of contents, an index and explicit cross-referencing. Requirement statements should be non-redundant where possible.

Traceable
The origin of each requirement should be clear, thus facilitating 'backward traceability' to previous decisions made, and 'forward traceability' to all documents 'spawned' from the requirements document.

The IEEE guide (IEEE, 1984) states that care should be taken to distinguish between the model for the application and the model for the software which is required to implement the application model. Further the guide recommends that whatever type of model is used it must be rigorously defined and kept within the domain of the requirements.

The model which is to be used as the basis for the specification is crucial, as it must integrate widely-differing types of information from users. Important characteristics that such a model (and its associated language) should possess (according to Flynn et al., 1986, Sowa 1984, Verheijen and Van Bekkum, 1982) are:

1. high level of abstraction :The model should be at the level of the users' views of the desired system, and should capture their concepts directly. It should not indiscriminately mix this high level information with information that is relevant to lower levels of the development process (for example, details concerning data representation or physical device implementation).

2. human-readable :The language in which the model is to be expressed will be used for validating the specification. That is, for presenting the specification to users for their views on its contents. Human understandability is thus the prime concern; and there is a need to avoid the well-known problems that users experience in trying to understanding formal requirements specification languages.

3. precise :A high-level specification language, for project scope agreement, delineating the system boundaries and naming major objects, rules and processes, is required. Further detail should be left to subsequent development phases. However, the language must be precisely defined, to reduce ambiguity, and to allow for formal integration and consistency checking methods to operate on its representational form.

4. specification completeness : It is important that the model captures all aspects of a specification, particularly the HCI specification and, for example, system timings for real-time applications.

5. mapping to later phases : A Requirements Definition phase will typically be followed by detailed analysis and design phases, and the model should therefore possess a structure suitable for mapping onto the later phases.

Thus the requirements document itself may have different roles and may have differing forms and content. None the less there are qualities which the documents (and the models contained within them) should possess.

Thus, there is a need for RE techniques which:

- support identification of requirements for generic products
- are capable of working alongside market analysis techniques
- support analysis of competitive products
- support predictions of future users and future use, and estimations of future usage
- support generic descriptions of typical users and groups of users
- support identification and description of current workpractices
- support identification of constraints such as cost, time and security
- support identification and specification of acceptance criteria

- support identification of organisational objectives, of key stakeholders and their objectives, and of key workgroups and their objectives
- support identification of work roles to be supported and why, and descriptions of each work role, and functional requirements to support each work role
- support identification and specification of quality attributes: usability, reliability, portability, performance, security, maintainability, acceptability and so on depending on the proposed system.
- support identification and specification of requirements for user documentation, requirements for training, requirements for user support
- support identification and description of human computer interface requirements
- encourage the writing of unambiguous statements
- encourage a complete specification to be written
- enable the development of verifiable (measurable) requirements statements
- encourage consistency when writing requirements statements and provides support for checking consistency
- encourage the development of modifiable requirements documents, for example, through indexing and cross referencing.
- enable traceability of requirements both backwards in the RE stage and forwards into design, that is traceability both of the requirements and of associated documents.
- encourage the following qualities in models developed: a high level of abstraction, human readable, precise, complete and has the ability to map to later phases

The discussion so far has focused on the RE process, on human communication, on knowledge development and on the form and content of the requirements document. The next section is concerned with the problem of managing the RE process, managing communication between people including knowledge development and managing the requirements document.

2.6. Management

In Chatzoglou and Macaulay, 1995, Requirements Engineering is described as a Project Managers Dilemma below is an extract from the paper:

"Consider the following scenario : you are asked to undertake the task of managing a new project development. You have not been involved in the 'concept' stage of the project, but as a professional Project Manager it is reasonable for you to be asked to manage the requirements stage of the project.

The remit you are given can be summarised as follows :

1. *You must get the requirements 'right'. We must build the product the user wants. Quality is all important.*
2. *Here is a document which describes the overall concept of the product. You will need to work out the detail.*

3. *You must get the requirements signed off by the marketing, production, user support and strategy managers and the customer authority.*
4. *You must complete in one month. You can have one senior and two junior analysts seconded part-time to the project, and a budget of £10,000."*

The Project Managers Dilemma is that it appears to be impossible to achieve the first three of the above given the resources stated in four. However, the Project Manager must develop a plan of how the requirements capture 'team' should proceed."

Chatzoglou and Macaulay report on research to identify those factors which affect the project plan and in particular which will enable the project manager to achieve the first three of the above using the minimum of resources.

They argue that in the Requirements Engineering (RE) process, productivity is about gathering 'enough' agreed[2] requirements in order to proceed to the design stage with the minimum cost, time and effort. Productivity in the RE process can be measured in terms of the number of people involved, the time taken in the RE process (elapsed time), the cost of the RE process, the effort in man-months and the amount of requirements gathered.

The purpose of their research was to identify those factors which affect productivity within RE and in addition to identify the relationship between those factors and the RE productivity measures.

Data from 107 projects was gathered from 70 different organisations within the UK. Respondents were project managers (51%), systems analysts/designers (30%) and consultants (19%) all were experienced in requirements. Respondents were asked to answer questions about the last project they were involved in. The high rate of response to questionnaires (48%) indicates that respondents feel strongly about the requirements stage of a project. Many feel that insufficient resources are given to RE. The following are representative quotations from respondents.

"*I suggest you consider, very seriously, that RE is a separate exercise - the IT strategy is determined by the RE process.*"

"*In general the client organisation under-appreciated the importance of the requirements specification stage*".

A project manager who worked for a large company and who participated in a large[3] project stated :

---

[2] 'Agreed requirements' means agreed between those people who have to 'sign off' the requirements before design commences, people such as those mentioned in 3 above.

[3] (the cost of the project was around $1M; the elapsed time to completion of the project was 5 years; more than 30 people were involved; the cost and elapsed time allocated to RE process was less than 5% of the total project)

".....The project (which is a large project), did not really have a properly identified RE stage. Many problems came from this!

If the project had included a proper RE stage the overall goals of the project would have been substantially reduced as it would have become clear that certain 'requirements' were very ill defined, and, that in order to fulfil them a great deal of work would have been required to properly characterise them.

Reducing the project goals would have substantially reduced design time and would have resulted in a product brought to market more timely. It is now clear that certain of the 'requirements' were of little value."

Chatzoglou and Macaulay, found that project management, project team members' attitude, and users' participation and communication with team members were very important for the success of the RE process in most of the projects. The key factors that affect productivity in the RE process are:

i) Team members' attitude towards the systems development process, including:

1. *Team members experience with system development and with establishing requirements;*
2. *The team's knowledge of the problem domain;*
3. *The team's commitment and persistence to the development of the specific system;*
4. *The team's anxiety and stress for the successful completion of the RE process;*

ii) Users' participation and attitude towards the system development process, including:

1. *User participation in the development process and their communication with team members;*
2. *User knowledge of the purpose of system development;*
3. *User motivation;*
4. *Conflicts between users;*
5. *User resistance to accept the development of the specific system;*

iii) Project management, including:

1. *Resources available for the completion of the RE process;*
2. *Techniques and tools employed during the RE process; and*
3. *Management style adopted.*

iv) Project characteristics, including:

1. *Project type (Software, Hardware, Systems);*
2. *Target users (Own use, External use);*
3. *Applicability (Bespoke, Generic);*
4. *Problem domain (Well Defined, Moderately Defined, Poorly Defined).*

v) Other

1. *Developers (Software. House, Industry, Academics, Consult.);*
2. *Information sources (Customer, User, Documentation, Marketeer etc.);*
3. *Targets set (On Time, In Budget, Best Quality etc.).*

Management of the RE process is difficult, because as their survey also found:

1. RE is an iterative process since in only 18% of the projects just one iteration was performed. In 32% of the projects two iterations occurred, while in 50% of the projects the RE process was completed in three or more iterations.

2. The elapsed time of the RE process usually represents more than 15% of the total elapsed time; however the cost of the RE process is 5-15% of the total cost

3. The failure of 35% of the projects to capture the necessary requirements is caused by:
    - Lack of time (61% of the projects);
    - Poor access to information (51% of the projects);
    - Insufficient manpower (22% of the projects); and
    - the cost (19% of the projects)[4].

The reasons given for changes in original plans were: lack of information, the need to validate and verify the information captured, assurance that everything has been done properly, and also, technology and market changes, user behaviour and inexperienced project managers.

They also found that the more time spent in the RE stage, the less time spent in the whole development process. Further the higher the cost of the RE stage the lower the cost of the whole development process. Thus the management of the RE process affects not only the RE stage itself but the whole of the project development.

Some of the issues raised above refer to the management of the RE process: the choice of the RE team, use of methodologies, resources available and management style. While other relate to knowledge development : access to users, communication with users and knowledge of the problem domain.

Other recent work is concerned with the management of the requirements document and traceability of it's content into design. For example RTM 1994, CARD (Ohnishi and Agusus, 1993), DOORS, 1995 and READS (Smith, 1993)

Thus RE techniques are needed which:

- support the management of the RE process
- are capable of producing cost estimates of the RE process
- are capable of modelling the RE process

[4] The sum of percentages is greater than 100% because more than one option could be chosen.

- support project planning specifically for RE projects
- are capable of fitting with IT Strategy techniques
- help the project manager identify the skills needed to complete the RE process
- help quantify the factors affecting productivity for a given project
- support traceability of requirements from the requirements document into design and later project stages
- support traceability of requirements from ‘concept’ through to the requirements document
- support the management of knowledge development
- support the management of human communication
- support the development of a ‘knowledge base’ about previous projects with respect to the RE process and project success.

2.7. Requirements for RE Techniques

The role of RE techniques can be summarised as being needed to support each of the areas shown in figure 8 below:

<table>
<tr><th colspan="2">Requirements Engineering Techniques</th></tr>
<tr><td rowspan="4">Management</td><td>The RE process</td></tr>
<tr><td>Human communication</td></tr>
<tr><td>Knowledge development</td></tr>
<tr><td>The requirements document</td></tr>
</table>

Figure 8 The role of Requirements Engineering techniques

The seventy requirements identified are listed below:

PROCESS
1. support articulation of the product concept
2. support problem analysis
3. support feasibility studies and cost-benefit analyses of options
4. support analysis and modelling
5. support documentation of requirements
6. support a systematic step by step approach
7. provide standardised ways of describing workproducts
8. provide procedures for maintaining workproducts

9. provide ways of assessing the quality of workproducts
10. enable identification of measures and measurement of the RE process
11. supports descriptions of effectiveness in RE terms
12. support analysis of opportunities for process improvement
13. provide automated support for the RE process

HUMAN COMMUNICATION TECHNIQUES WHICH:

14. provide guidance on interviewing users
15. provide guidance on the design and use of questionnaires
16. provide guidance on conducting observations of users
17. support identification of various viewpoints
18. support reconciliation of viewpoints
19. support the user in reviewing models developed
20. support users in analysing their own problems and identifying the need for change
21. support construction of appropriate requirements teams
22. support identification of stakeholders
23. support the development of a 'shared meaning' of the system being specified
24. encourage intuition, imagination and common sense among participants
25. support communication between people from a diversity of backgrounds
26. support facilitated meetings with predefined agendas and problem solving strategies
27. support the development of listening skills among participants

KNOWLEDGE DEVELOPMENT TECHNIQUES WHICH SUPPORT:

28. Relevant structures on the users' present work
29. Visions and design proposals
30. Overviews of technological options
31. Concrete experience with the users' present work
32. Concrete experience with the new system
33. Concrete experience with technological options
34. Knowledge of the current market and insights into future market changes
35. Knowledge of current and proposed competitor products
36. Knowledge of company strategy and likely future developments
37. Knowledge of the current organisation and the potential for change
38. Knowledge of government policy and planned change

REQUIREMENTS DOCUMENTATION TECHNIQUES WHICH:

39. support identification of requirements for generic products
40. are capable of working alongside market analysis techniques
41. support analysis of competitive products
42. support predictions of future users and future use, and estimations of future usage
43. support generic descriptions of typical users and groups of users
44. support identification and description of current workpractices
45. support identification of constraints such as cost, time and security
46. support identification and specification of acceptance criteria
47. support identification of organisational objectives, of key stakeholders and their objectives, and of key workgroups and their objectives

48. support identification of work roles to be supported and why, and descriptions of each work role, and functional requirements to support each work role
49. support identification and specification of quality attributes: usability, reliability, portability, performance, security, maintainability, acceptability and so on depending on the proposed system.
50. support identification and specification of requirements for user documentation, requirements for training, requirements for user support
51. support identification and description of human computer interface requirements
52. encourage the writing of unambiguous statements
53. encourage a complete specification to be written
54. enable the development of verifiable (measurable) requirements statements
55. encourage consistency when writing requirements statements and provides support for checking consistency
56. encourage the development of modifiable requirements documents, for example, through indexing and cross referencing.
57. enable traceability of requirements both backwards in the RE stage and forwards into design, that is traceability both of the requirements and of associated documents.
58. encourage the following qualities in models developed: a high level of abstraction, human readable, precise, complete and has the ability to map to later phases

MANAGEMENT TECHNIQUES WHICH:

59. support the management of the RE process
60. are capable of producing cost estimates of the RE process
61. are capable of modelling the RE process
62. support project planning specifically for RE projects
63. are capable of fitting with IT Strategy techniques
64. help the project manager identify the skills needed to complete the RE process
65. help quantify the factors affecting productivity for a given project
66. support traceability of requirements from the requirements document into design and later project stages
67. support traceability of requirements from 'concept' through to the requirements document
68. support the management of knowledge development
69. support the management of human communication
70. support the development of a 'knowledge base' about previous project with respect to the RE and project success.

This list of requirements for RE techniques is offered as an attempt to identify what techniques are needed and why. The list is not comprehensive, some of the requirements are at different levels of detail, none the less the author believes that by developing a 'wish list' of what is needed the requirements engineer will be better able to evaluate the techniques which are available.

2.8 The contribution of various approaches

In this section the author refers back to the approaches to the problem of requirements described briefly in section 1.9 and considers how each approach contributes to the requirements for RE techniques, that is, the ‘wish list’. Figure 9 below presents a mapping from the ‘wish list’ to the approaches. Each ‘X’ means that the author can identify a contribution from a particular approach to an item in the ‘wish list’.

Figure 9 The contribution of the various approaches to the requirements for RE techniques in the ‘wish list’

Figure 9 serves to illustrate the strengths of each of the nine approaches, for example, structured analysis makes a contribution to most areas but is weak on knowledge development. Participatory Design approaches are strong in human communication and knowledge development but weak in almost every other area.

Figure 9 also serves to reinforce the view that no single approach to requirements will provide the requirements engineer with all the tools and techniques needed.

It is not appropriate for a requirements engineering book of this type to cover all possible techniques from the above approaches. Indeed an encyclopedia would be needed. Thus the next three chapters focus on particular areas within the matrix of figure 9. The areas were chosen such that they represent particular groupings of techniques which address particular problems.

In the introduction to this chapter it was suggested that the objective of the requirements engineering process was to specify a system which was ultimately proven to be successful. The purpose of the next three chapters is to present examples of common causes of system failure and to suggest candidate techniques which could be used to prevent the cause of failure. Each chapter focusses on a different type of failure (as defined by Lyytinen and Hirschheim, 1987). Chapter three focusses on expectation failure, chapter four on process failure and chapter five on interaction failure.

1. Expectation failure can occur when insufficient attention is paid to the social and organisational context. More specifically when there is a failure to realise that the goals of a system are defined within the total context of an organisation and its social and political environment and not just in relation to technology. (Robinson , 1994, Eason, 1987)

2. Process failure can occur when different interest groups do not communicate effectively with each other, each seeking to exert power and influence over the other. (Gasson, 1995, Bjorn-Anderson, 1986)

3. Interaction failure can occur when requirements engineers and designers do not fully understand the work of users. (Greenbaum and Kyng, 1991)

Figure 10 below shows the areas of the matrix which the next three chapters address:

Figure 10 The areas addressed by the next three chapters.

Figure 11 shows the relationship between the type of failure and the chapter title.

Type of Failure	Example of cause of failure	Requirements from wish list	Candidate techniques	Chapter title
Expectation failure	social and organisational issues not addressed	1, 2, 3	SSM ETHICS UCSD	Organisational Approaches Chapter 3
Process failure	conflicts between interest groups	21 to 27	CRC QFD JAD	Group Session Approaches Chapter 4
Interaction failure	lack of understanding of users work	28 to 33	Future workshop Cooperative prototyping, Cooperative evaluation	Interactive Approaches Chapter 5

Figure 11 : Mapping from the type of failure to the chapter title

2.9 Summary

In this chapter three types of system failure were discussed, and five possible causes identified. The causes were:

- lack of a systematic RE process,
- poor communication between people
- poor management of people and resources.
- lack of appropriate knowledge or shared understanding and
- inappropriate, incomplete or inaccurate documentation.

Each cause was discussed in some detail and the need for various requirements engineering techniques identified. The chapter concluded with a 'wish list' of seventy techniques and a matrix showing a mapping from the 'wish list' to the approaches discussed in chapter one. Specific areas within the matrix have been chosen as the focus for chapters three, four and five.

The next chapter, chapter three, concentrates on one cause of expectation failure and presents a number of requirements techniques which could be used to avoid such a cause of failure.