New copy

CRANFIELD UNIVERSITY

MICHAEL INCE

INFORMATION SYSTEMS – A METHODOLOGY FOR CAPTURING USER REQUIREMENTS

SCHOOL OF INDUSTRIAL AND MANUFACTURING SCIENCE

PhD THESIS

CRANFIELD UNIVERSITY

SCHOOL OF INDUSTRIAL AND MANUFACTURING SCIENCE

PhD THESIS

Academic Year 1999-2000

MICHAEL INCE

Information Systems – A Methodology for Capturing User Requirements

Supervisor: Professor John. M. Kay

April 2000

This thesis is submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy

ABSTRACT

Systems development has been a topic of discussion for many years, with the growth of the Information Technology (IT) industry. Many organisations use systems to allow them to have more control over the processes that are carried out on a daily basis. The Systems Development Life Cycle (SDLC) has had a major impact on the way that systems are developed. It provides a structure that begins with identifying the feasibility of the project to be completed, and guides the project through implementation and maintenance of the system. It has been suggested that the lack of user involvement during this development process is a major cause of this failure in the systems that are developed. This research addresses this problem by reviewing the most popular methods and examining the concept of user involvement. An industrial perspective also assists in generating key issues in systems development. These, together with the lessons learned during the literature review form the basis of the development of a new approach.

The new approach proposed in this thesis (URCAM, User requirements Capturing Method) comprises a set of stages and phases that a developer can use to carry out the process of capturing user requirements. While learning from the weaknesses of the existing methods, the proposed method considers their strengths.

URCAM was evaluated during a systems development project at BICC General Cables. The results of the evaluation suggested that the method would produce higher quality information than verbal communication. However, the major strengths of the method are that it offers the developer a guide that they could follow from identifying the types of users who would be involved, to the sign-off, of the final specification. URCAM forms are also presented, which offer the user an opportunity to write down information. The results suggest that this method helps the user to feel more involved and increases their willingness to use the implemented system.

ACKNOWLEDGEMENTS

I would like to thank a number of people that have contributed to this research. There are too many to mention them all by name. However, I would like to mention a few specifically.

My sincere thanks are offered to my supervisor Professor John. M. Kay who has been my academic guide throughout my thesis. I would like to thank him for his continued help, support, encouragement and commitment to my studies.

I would like to thank all of the staff, Doctoral students, Masters students, the friends that I have made while I have been at Cranfield and the friends that I had before I came to Cranfield for their friendship and their encouragement.

I would like to thank all of the staff at BICC General Cables, especially Reza Rahmani-Torkman, Daniella Wright and Newton Matupi and the rest of the MSE (Manufacturing Systems Engineering) department for their help in giving me an industrial awareness of the important issues in IT.

Finally, my gratitude is offered to my Father and Mother and the rest of my family for their help in getting me where I am today and where I will go in the future.

I thank you all.

LIST OF CONTENTS

CHAPTER 7
DISCUSSION AND CONCLUSIONS

LIST OF APPENDICES

LIST OF FIGURES

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Chapter 6

Appendix A

LIST OF TABLES

GLOSSARY OF TERMS

ADDISSA	Architectural Design of Information Systems based on Structural Analysis
ADS	Accurately Defined System
CBA	Cost Benefit Analysis
CIE	Customer Interface Engineer
CIMOSA	Computer Integrated Manufacturing Open Systems Architecture
DFD	Data Flow Diagram
DSDM	Dynamic Systems Development Method
ER	Entity Relationship
ETHICS	Effective Technical and Human Implementation of Computer Based Systems
EUC	End User Computing
G	Goal
GERAM	Generic Enterprise Reference Architecture and Methodology
GRAI	Graphical Results and Activities Interrelated
IDEF0	ICAM Definition 0
IRR	Internal Rate of Return
IS	Information Systems
ISD	Information Systems Development
ISM	Information System Methodology
IT	Information Technology
JAD	Joint Application Development
MoSCoW	Must have, Should have, Could have, Would have
NIMSAD	Normative Information Model-Based Systems Analysis and Design
NPV	Net Present Value
OOA	Object Oriented Analysis
PSL	Problem Solving Language
RA	Requirements Analysis
RAD	Rapid Application Development
RADD	Rapid Application Data Development
RC	Requirements Chunk
RE	Requirements Engineering
ROI	Return On Investment
RS	Requirements Specification
RSL	Requirements Statement Language
SARA	Systems Architect Apprentice
SB	Structured Brainstorming
Sc	Scenario
SD	Systems Design
SDLC	Systems Development Life Cycle
SDM	Systems Design Methodologies
SOP	Study Organisation and Plan
SOS	System for Supporting and Selection process
SSADM	Structured Systems Analysis and Design Method
SVL	Systems Verification Language
U/DIP	User/Developer Interactive Process

UI	User Involvement
URCAM	User Requirements Capturing Method
YSM	Yourdon Systems Method

CHAPTER 1

INTRODUCTION

1.0 Introduction

For some thirty years, organisations have been developing Information Systems (IS) (Edwards et al, 1995). Gough (1997) states that there is no evidence to suggest that this trend of developing IS is slowing down, if anything the pace of innovation is quickening. However, even though this technological pace is getting faster, Mitchell (1997) states that the successfully implemented Information Technology (IT) project is one of the most elusive creatures to be found in the modern day enterprise. He also states that many projects are still failing miserably and of those that do limp into existence many are late or over budget. Some do not achieve the functions for which they were originally commissioned. While others still match the specifications, but fail to match the users needs because the system was badly specified in the first place.

In order to learn from the mistakes that have been made in the past, the causes of the problems should be identified and the solutions should be created to deal with those problems. One of the causes that have been cited is the interaction between users and developers during the process of systems development. Zarley (1989) argues that as users become more proficient in automating business tasks, the relationship between users and the IS department must change. This suggests that the communication between the two parties needs to be improved in some way. However, this is not the only perspective found. Bekker and Vermeeren (1993) found that designers (developers) felt that they had a lack of information about users and that they experienced obstacles to user involvement in the design process. Gouguen and Linde (1993) state that a major barrier to achieving high quality requirements within the requirements process in organisations is the user-developer culture gap. Flynn and Warhurst (1994) state that the type of problem that arises is that users and developers use different languages for communicating about the domain.

Colter (1984) states that while early structured concepts and methods concentrated on improving coding practices, structured design techniques soon followed in response to a realisation that a well structured implementation of poorly designed systems yielded few benefits. The idea of structure presents another area of interest in systems development. The Systems Development Life Cycle (SDLC) as cited by Folkes et al (1992) encompasses all of the main stages in developing IS. This has evolved into the development of a vast number of systems development methodologies. Clarke and Moriera (1999) state that a requirements analysis method enables people to understand and to structure the information collected during requirements capture, to identify inconsistencies, omissions, ambiguities in the requirements, to model the observable behaviour of the proposed system and to analyse the resulting model to ensure that the requirements are accurately represented. Fitzgerald (1996) estimates between 300 and 1000 different published methodologies. Olerup (1991) states that the difference in methods usually occurs in the form of philosophy, objectives and techniques and may also differ fundamentally in paradigm, some having a 'hard' scientific and rationalistic approach, and others a 'soft' human-oriented one. He states that the 'hard' approach is characterised by a subdivision of the development process. The complex development process is broken down into categories of analysis of requirements, design of a solution, and implementation of that solution. Mumford (1983) states that the 'softer' approach is advocated by a more user-centred approach where a more social perspective is achieved.

Barki and Hartwick (1994) state that in IS, the term "user involvement" has generally referred to a series of activities or behaviours performed by potential users or their representatives during the systems development process. In contrast, the term "user involvement" has been used in other fields to describe a subjective psychological state reflecting the importance and personal relevance of an issue (e.g Sherif et al, 1965), or of an individuals job (e.g. Lawler and Hall, 1970). Therefore, in other fields, involvement reflects an individual's beliefs or feelings concerning some object. Therefore, to align work in IS with that of other disciplines, Barki and Hartwick (1989) make two recommendations: (1) to use the term "user participation" instead of "user involvement" when referring to their assignments, activities, and behaviours that users or their representatives perform during the systems development process, and (2) to use the term "user involvement" to refer to a subjective psychological state reflecting the importance and personal relevance that a user attaches to a given system. However, Clavadetscher (1998) states that user involvement is the key to success in software development. Engler (1996) states that of a survey of 300 IS executives, the results suggest that the lack of user involvement is the main reason IS projects fail. These statements present user involvement where the papers have not distinguished their separation of user involvement and user participation. Therefore, in order to appeal to a wider audience, this study has decided to adopt the definition where user involvement as stated by Barki and Hartwick (1994) is referred to as a series of activities or behaviours performed by potential users or their representatives during the systems development process. Therefore, where Barki and Hartwick use "user participation" the same meaning is used for "user involvement" in this research.

1.1 Objectives

The introduction to this thesis has shown that various technologies and techniques have evolved, which have assisted the systems development process. However, there are still major problems with the possibility of achieving IS success. The overall aim of this thesis is to review the problems associated with IS that relate to user involvement, and to produce a method of capturing user requirements during the systems development process. The method should adopt the strengths from existing systems development methodologies and requirements from industry. The method should also contain a set of phases and stages that can be used to guide a developer through the process of capturing user requirements, and have a template that can be used to capture the requirements to give the user a more important role within the systems development process. The method should also be flexible enough to be adapted to different situations, for example data collection and analysis.

The objectives of the research described in this thesis are as follows:

1. Identify and compare existing systems development methodologies and methods designed to address user requirements. Discuss their strengths and weaknesses in relation to the extent and appropriateness of user involvement within their structure.
2. Investigate the validity of the belief that user involvement is a major problem that affects systems success or failure.
3. Identify the requirements for a new method of capturing user requirements from both parties (users and developers).
4. Develop a method for capturing user requirements. The method should consider the strengths offered by existing methods and the requirements stated by users and developers in industry. Then evaluate the method and identify what the results of the research indicate.

1.2 Structure of the thesis

Chapter 1	**INTRODUCTION** Identifies the research problem and states the overall aim and objectives of the research.
Chapter 2	**LITERATURE REVIEW** Reviews the literature concerning the Systems Development Life Cycle, structured methods and discusses additional literature relating to the solutions to the problems concerning user involvement in systems development.
Chapter 3	**RESEARCH METHODOLOGY** Discusses the options available for research and selects appropriate methods to develop a suitable research methodology.
Chapter 4	**SURVEY OF USERS AND DEVELOPERS** Presents the results of a survey of users and developers carried out as a field experiment and data collection exercise. Identifying the problems and the requirements from industry in relation to systems development and user involvement.
Chapter 5	**THE DEVELOPMENT OF URCAM** Explains the process of developing the specification of the new method for capturing user requirements and presents the new method.
Chapter 6	**THE EVALUATION OF URCAM** Evaluates the model using an industrial survey where methods suggested in the research methodology are used.

Chapter 7 DISCUSSION

A discussion of the research with respect to the research objectives is completed in this chapter. Conclusions are also drawn together with stating how the research has contributed to knowledge, its limitations and makes recommendation for further research.

CHAPTER 2

LITERATURE REVIEW

2.0 Introduction

This chapter reviews the literature that relates to Information Systems (IS) and user involvement.

It begins with a definition of IS and then discusses how the area of IS development has evolved into a more structured process. The work carried out by a collection of authors who have studied the development of structured methodologies is discussed.

The problems with IS development are identified within this chapter. These problems highlight the growth of interest in IS and why they succeed or fail. These problems included factors, which include a lack of user involvement. Many authors including Cheney and Dickson (1982), Yaverbaum (1988), Reubenstein et al (1991), Koh et al (1996), Mumford (1997) and Taylor (1998) have argued that the user is an important consideration within the development of IS, and their role requires more attention than they have been given in the past.

Systems development methodologies were then reviewed to identify whether they considered users in their development process. The review concluded that there was room for improvement. Therefore, the methods that have been developed to overcome the problems that were being faced were reviewed. These methods had not been very well cited and are relatively new. The strengths and weaknesses of both systems development methods and methods developed to capture user requirements were generated. The final stage was to identify that there was a need for a new approach. The weaknesses identified from the literature were used as the basis for this argument.

2.1 Information Systems Development

2.1.1 Background

Lucas (1992) states that an information system is a computerised set of organised procedures that, when executed, provides information to support processes, decision making and control in the organisation. Sprague and McNurlin (1993) suggest that the mission of an information system in an organisation should be to improve the performance of people in organisations through the use of information technology. However, they also state that this mission for the IS function is still fraught with difficulties, such as how to define and measure performance, but it is not limited or short-sighted. Avison and Fitgerald (1995) state that the 'system' part of 'information system' represents a way of seeing the set of interacting components, such as:

People (for example, developers and business users)
Objects (for example, computer hardware devices)
Procedures (for example, those suggested in an IS development methodology)

They state that an information system in an organisation provides facts useful to its members and clients, who should help it operate effectively. However, Gorry and Morton (1971) developed a framework that acts as a starting point for systems development. It considers three major aspects of IS development: Operational Control, Management Control and Strategic Planning. These three considerations are used to form the relationship to create the system, taking into account the planning of the process, the operational requirements and the management decision making. However, to form a boundary around the research it is important to acknowledge that close attention is paid to the relationship between IS and how it is developed, the problems with the way that they are developed and how the development process can be improved.

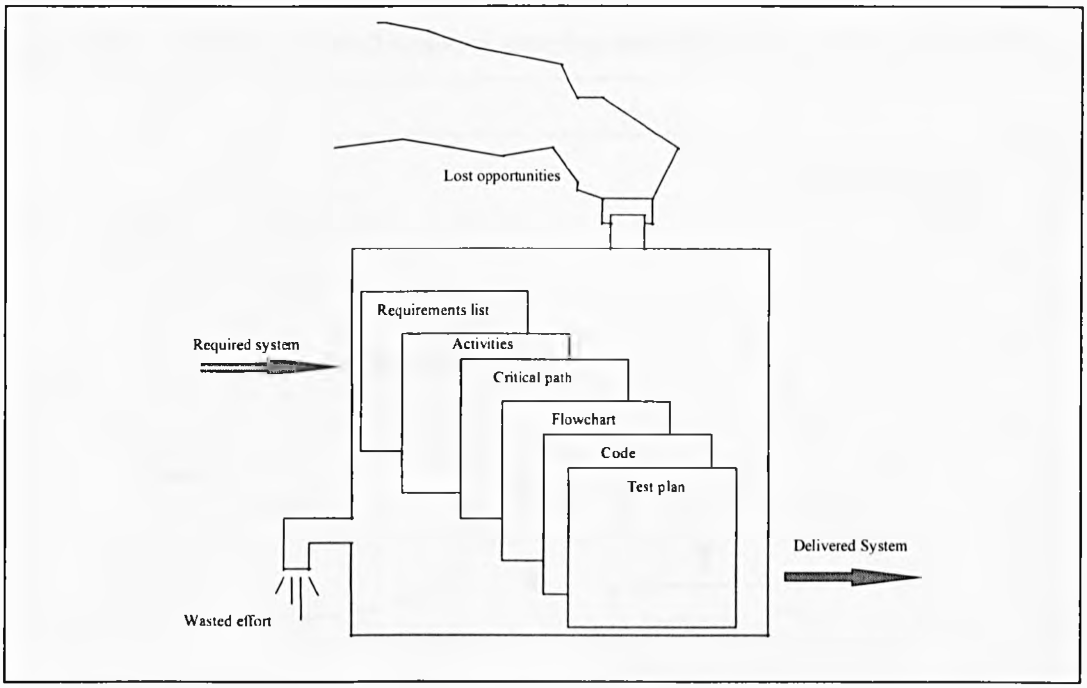
Definition:

After reviewing the work of Lucas (1992), Sprague and McNurlin (1993), Avison and Fitzgerald (1995) and Gorry and Morton (1971) it is suggested that the definition that the study will adopt is that an Information System (IS) contains people, objects and procedures. It provides support and control to improve strategic planning, operational control and management control. The main reason for this suggestion is that the different authors have all addressed some relevant factors that are presented in the literature. These factors relate to IS development and to choose one definition may not generate an adequate information boundary for completing the goals of the study.

2.1.2 The Systems Development Life Cycle (SDLC)

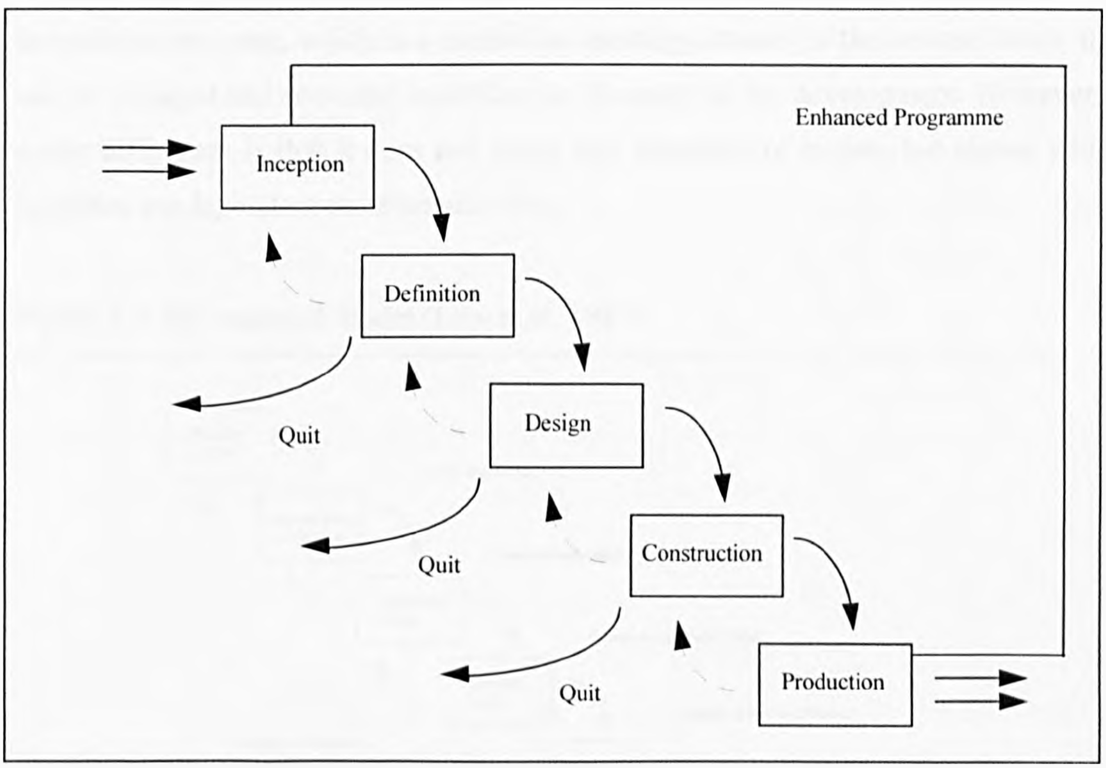
The Systems Development Life Cycle is used to solve systems problems, by formalising the process of problem solving. This process will ultimately analyse the problem and then go through an implementation process, which should benefit the company that requires the new system. The traditional method of completing a systems development project was by using techniques generated from various disciplines to produce a desired outcome. A typical method of development that has been used is called the traditional approach. Figure 2.1 demonstrates an example of the traditional approach.

Figure 2.1: Traditional project development process (Folkes et al, 1992)



Folkes et al (1992) state that firstly the traditional methods did not satisfy the business needs of a company and secondly that it took too long and had too many compromises to make it acceptable. They state that a solution to this problem would be to introduce some form of phased approach, where each phase is controlled by the techniques to be used and the documentation to be produced. The use of this type of method would mean that at each stage or phase there would be the possibility of choosing one of three options once the stage has been completed. The three options may be as follows: (1) to continue to the next stage, (2) to abandon the project or to (3) backtrack and change specified requirements or alter decisions made within that particular stage. This is a good way to operate as there is always room for improvement. The method mentioned here is usually referred to as some form of Waterfall model. This is shown in figure 2.2. A more detailed version is shown in figure 2.3.

Figure 2.2: A typical waterfall systems development life cycle (Folkes et al, 1992)



The different stages in figure 2.2 are defined as:

- Inception - initial studies and feasibility study
- Definition - of functional, performance and security requirements of system; of necessary activities and data (i.e. includes analysis)
- Design - of business options, activities, data, interfaces
- Construction - of code, test procedures and interfaces, build database; systems testing
- Production - implementation; post-implementation reviews; optimisation; repair and enhancement

The main difference between the models presented by Law et al (1987) (see figure 2.4) and Folkes et al (1992) (see figure 2.1 and 2.2) is that Law et al's (1987) does not present a feedback loop at every stage. Another difference is that Law et al's (1987) includes prototyping, which is a method of creating a model of the desired result; this can be changed and recreated according to the stage in the development. However, a major difference is that it does not imply any sequence of events, but shows which activities are dependent on other activities.

Figure 2.3 The waterfall model (Law et al, 1987)

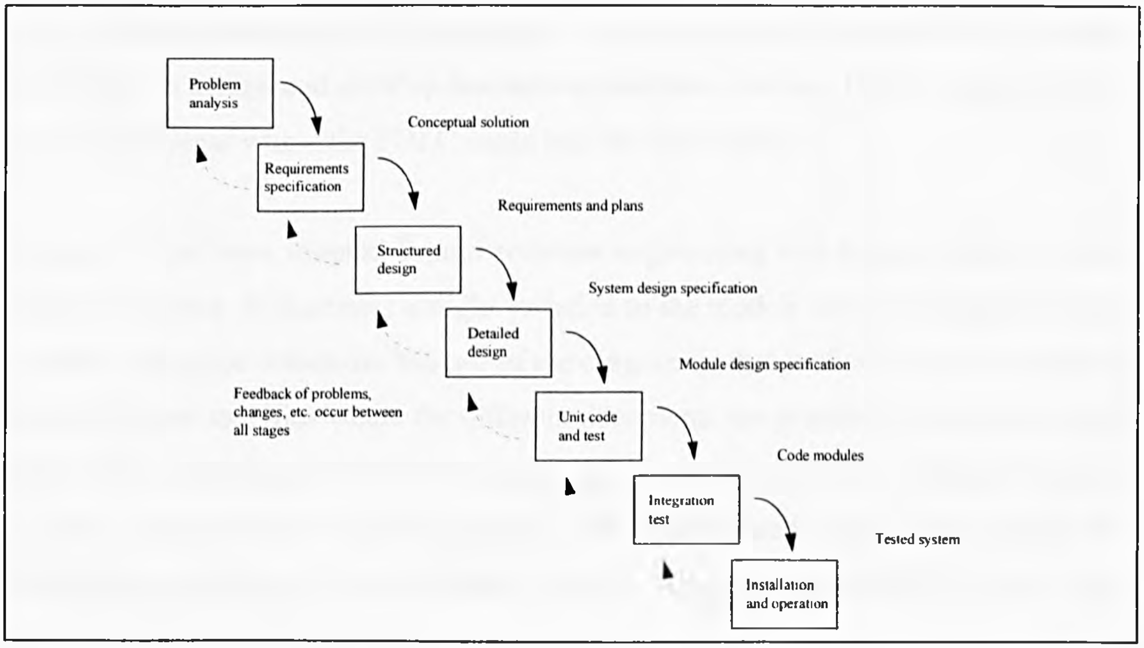


Figure 2.4 Refined model of evolutionary systems development (Law et al, 1987)

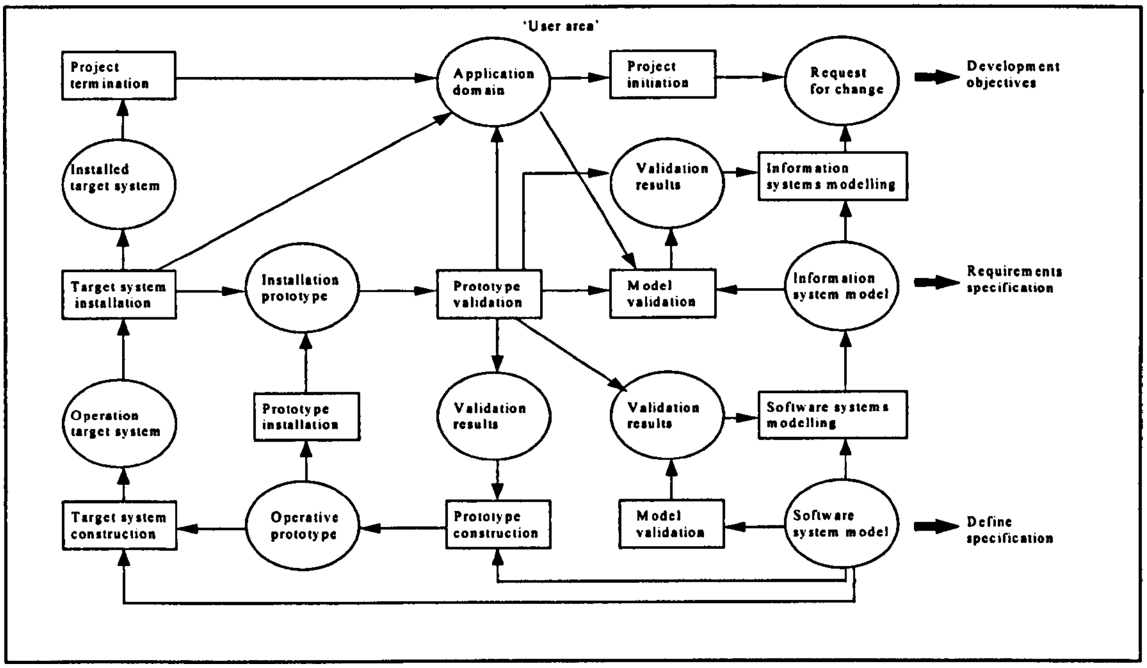
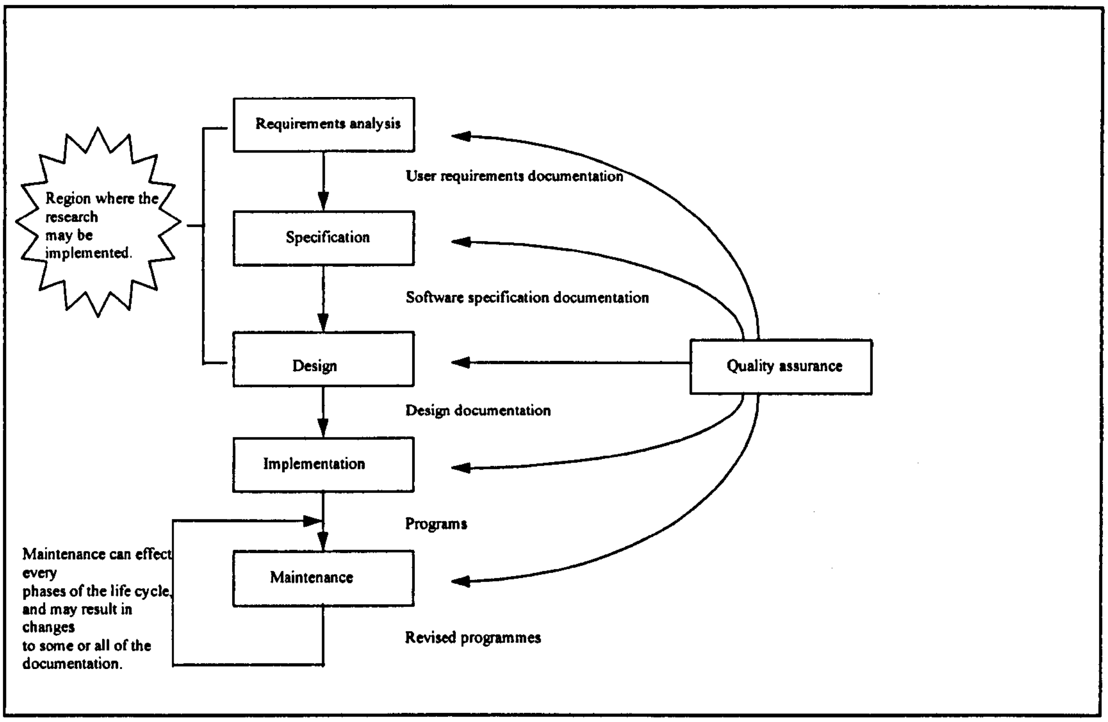


Figure 2.4 appears to be quite detailed. However, much can be learned from this approach. The main element is that the model can be produced to encompass all of the attributes that the conventional model has; it can also add further properties, such as prototyping and other suitable dependent activities that are required to make the new system a success. The SDLC has been used in literature to identify several problems with IS. Different experts have used the SDLC in case studies. A few examples include Couger (1985) who used the SDLC to compare the costs associated with life cycle phases for first and third generation computer systems. McKeen (1983) utilised the SDLC to design and develop business application systems. These examples show a variety of areas where the SDLC could and has been useful.

Figure 2.5 has been adopted from a software engineering text book produced by the Open University. It illustrates a slight variation to the models shown previously in this chapter. The main reason for the use of the diagram is that it shows a later version of the SDLC and specifies where the different documents are produced. It must be stated that various stages occur within the design and requirements phases. Using a structure at this stage should make this process less complicated, since there would be guidelines and stages to work towards. Once it is carried out correctly, there is the possibility of achieving an accurate specifications document which should lead to a system that contains the required functionality.

Figure 2.5: Research positioned within the SDLC (Adapted from Open University, 1994)



2.1.3 Structured methodologies used to develop Information Systems

In a survey conducted by Necco et al (1987), approximately two-thirds of a 97-firm sample used traditional IS development approaches (the SDLC and structured methodologies). In a more recent survey by Russo et al (1995), over three-quarters (76%) of a 133-firm sample reported using an SDLC-based approach to systems development. A selection of individually used methodologies have been described in Appendix A (Review of systems development methods), where it is possible to observe the various attributes that are available from the methodologies.

Nielsen (1989) states that practitioners are faced with numerous methodologies. Since one cause of development failure is the use of inappropriate or inadequate methodologies, it is important that practitioners choose the correct methodologies or techniques. Edwards et al (1989) conducted a study on a Structured Systems Analysis and Design Method (SSADM) selection in the UK. It found that most of the organisations had adopted SSADM because it was the official methodology for government agencies, and that only 11% of the respondents had used a formal methodology before adopting SSADM.

Heichler and Cafasso (1995) and Keyes (1992), state that technologies, such as fourth generation languages, application generators and local area networks, have changed the way systems can be developed, facilitating rapid, non-linear and more user-centred development. Wynekoop and Russo (1997) have classified over a hundred research papers, which are represented in Table 2.1 by research method and purpose. The sample was arrived at by identifying papers addressing Systems Development Methodologies (SDMs) (Culnan 1986; Straub et al, 1994)

Table 2.1: Classified research: number of studies in each category (Wynekoop and Russo, 1997)

	Research Question				
Research method	**Selection and adaptation**	**Evaluation**	**Use**	**Understand/ Describe**	**Totals**
Normative	0	1	0	63	64
Lab	0	1	0	0	1
Field	3	10	5	0	18
Survey	1	6	9	4	20
Case	3	5	3	0	11
Action	1	4	0	0	5
Practice	0	1	3	0	4
Interpretive	1	1	1	0	3
Descriptive	0	1	0	0	1
Totals	9	30	21	67	127

Table 2.1 relates to Table 2.2, which gives more detail to the research methods that were used in the study by Wynekoop and Russo (1997). Normative research dominates, accounting for over half of the work that has been classified. Wynekoop and Russo (1997) state that over a third of the papers in this category are frameworks for the selection, comparison or evaluation of SDMs. They are conceptual discussions of how SDMs should be selected, or evaluations on paper of whether or not a SDM has features that the author identifies as important.

The methodologies that were included in the study are shown in Table 2.3. The 'other' category includes SDMs addressed in fewer than five papers, including custom-developed SDMs and a methodological development. The 'unspecified' category includes papers that failed to identify SDM under study.

Table 2.2: Summary of research methods (Wynekoop and Russo, 1997)

Research Method	Description	Use/Example
Normative writings	Concept development not based on empiricism or theoretical grounding, but on the author's speculations or opinion. Descriptions include no interpretation, but are presented as objective accounts.	Frameworks, prescriptive writings. Theory building
Laboratory research	Treatments and independent variables are conducted in an artificial setting	Hypothesis testing; simulations
Surveys	Test hypotheses on data systematically collected from a known population via questionnaire administered in writing or orally	Study large samples; hypothesis development or testing; description
Field inquires	In field studies organisational changes are studied ex post facto based on recall and self-report with no variable manipulation. In field experiments, independent variables are controlled	Post hoc or real-time study of processes and outcomes; hypothesis formation
Action research	Research participates in the intervention or action studied. Researcher biases explicitly identified	Theory development and testing
Interpretive research	Attempt to understand a phenomenon, by studying it in its natural context from participants' perspectives. No controls, prior theories or attempt to generalise. Includes case studies and action research from this perspective	Achieve deep understanding of a phenomenon in one context, which may bring insight into others
Descriptive research	Interpretive research studying literature or past research or events; in-depth review of the phenomenon	Understanding a phenomenon
Practice descriptions	Descriptions of experiences by a practitioner, with implicit biases. No prior research intent	Descriptions of practices that failed in one company

Reviews carried out by Longworth (1985) and Bubenko (1986) concluded that there are at least 100 commercially available methodologies quite apart from the hundreds that are in use within companies as in-house aids. However, Lubars et al (1992), Southwell (1993) and Hopker (1994) state that there are a wide range of methodologies, but the number of projects where a methodology is used during the development process remains low.

It has been suggested by Loenard-Barton (1987) that experienced developers are more likely to use a methodology, since they would be aware of the benefits. Fitzgerald (1997) also carried out research that showed the relationship between developer experience and methodology usage. It has been suggested that some of the literature in this field states that the use of a methodology is more popular with less experienced developers because of the uncertainty in the development exercise. This conflicts with Leonard-Barton's statement, which may suggest that there is more than one school of thought in this area. Figure 2.6 shows a diagram of how Fitgerald's (1997) study identified this relationship. It is stated that educational exposure to methodologies might predispose inexperienced developers more towards a formalised methodological approach. Also inexperienced developers could find that a methodology provides a useful template for the development process.

Figure 2.6: Relationship between developer experience and methodology use (adapted from Fitzgerald, 1997)

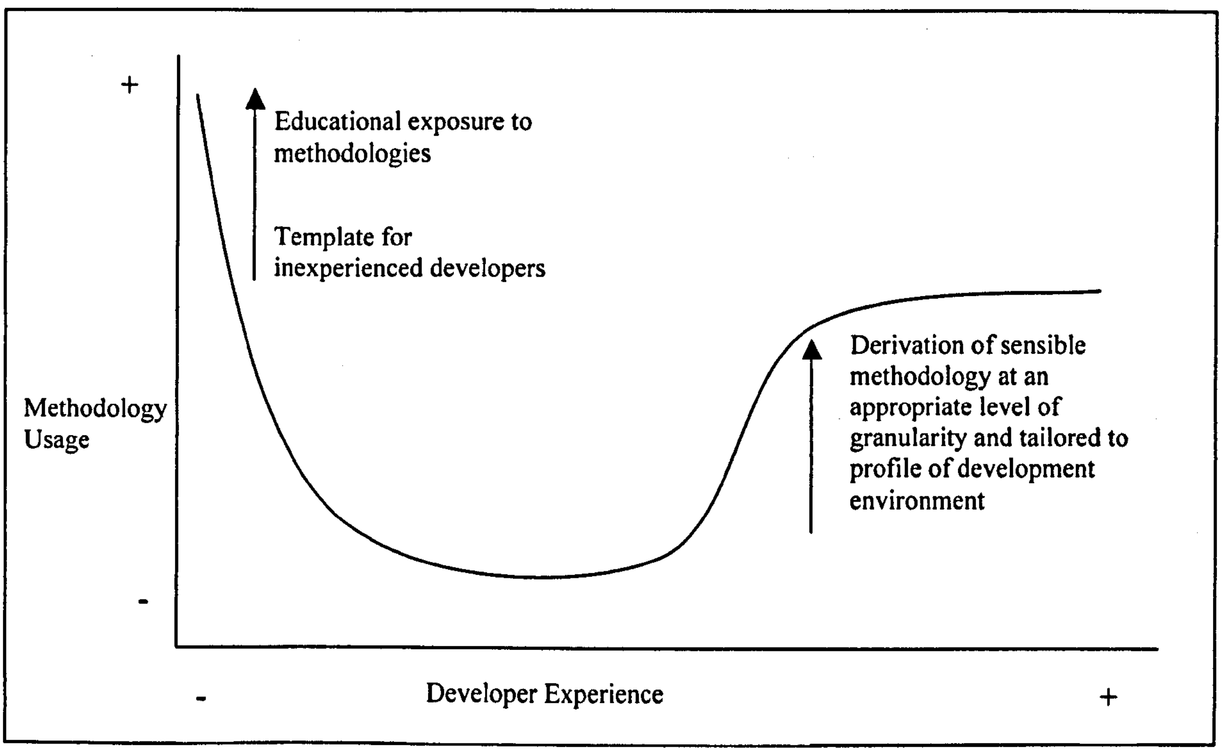


Table 2.3: Methods used in the study by Wynekoop and Russo (1997)

Method Abbreviation	Method name	No of studies
O	Other	16
PROT	Prototyping	11
SASD	Structured Analysis and Structured Design	11
TECH	Techniques (parts of SDMs)	9
SDLC	Systems Development Life Cycle	9
SSADM	Structured Systems Analysis and Design Method	9
OO	Object Oriented	6
UN	SDM not specified	6
JSD	Jackson Structured Design	5
IE	Information Engineering	4

2.1.4 Review of systems development methodologies

This section discusses the review of the methodologies identified for designing IS (see appendix A). The first question that this discussion answers is 'are the methodologies that were reviewed considering user involvement?' Table 2.4 identifies the strengths and the weakness as presented by each of the methods that relate to capturing user requirements. Table 2.5 then goes on to compare the methods, identifying whether they consider various factors that relate to user involvement.

Weaver (1993), Ince (1996) and Longworth (1992) cited research relating to the structure used in SSADM. From their research, it can be observed that requirements analysis and requirements specification are documented stages within the development process. This would suggest that the developers of SSADM have directly or indirectly input an area within their methodology where involving the user could possibly benefit the development of the final system. However, 'does SSADM state how the interaction between the developer and the user should take place?'

According to the research stated by the three authors, Weaver (1993), Ince (1996) and Longwoth (1992), there is no specific method within SSADM for involving the users.

Carter (1997) and Stapleton (1997) reviewed Dynamic Systems Development Method (DSDM). Their work suggests that DSDM has 9 principles, where the first is that 'user involvement is imperative'. Rapid Application Data Development (RADD), Rapid Application Development (RAD) and Joint Application Development (JAD) present a step towards finding a solution to the problems caused by a lack of user involvement. These methods (RADD, RAD and JAD) use workshops where users and developers occupy the same room. The users and developers will carry out the requirements capturing process within this room.

Vernadat (1996) identified the cycle used within Generic Enterprise Reference Architecture and Methodology (GERAM). The cycle includes a requirements stage, which suggests that there is room for the user to get involved. Kosanke (1995) identified the requirements definition and specification models within Computer Integrated Manufacturing Open Systems Architecture (CIMOSA). Therefore, the answer to the question ' are methodologies considering the user? up to this point the answer is 'yes'.

Domeingts (1985), McCarthy et al (1994), Wu (1994) and Chen (1997) presented Graphical Results and Activities Interrelated (GRAI) method as a decision making rather than a development tool. However, it is not clear were the user fits into the process. This is very similar to ICAM Definition 0 (IDEF0). Busby et al (1993), Cantemessa et al (1998), Komi (1995) and Wu (1994) have all cited work on IDEF0 and their work suggests that the user is a major factor in the phases involved. They state that the As-Is model and the Should-Be model need to be built. However, a requirements specification or analysis phase is not clearly stated. However, Cantemessa (1998) does suggest that there is a great deal of importance placed on people and their involvement in IDEF0.

The work of Avison and Fitzgerald (1995) and Yourdon (1993) on Yourdon Systems Method (YSM) suggests that the most important factor to consider during the systems essential model (a model within the YSM lifecycle) is to determine what the user will need. Their work suggests that the basis of YSM was that the system should be built using a top-down approach. The latest version according to Yourdon (1993) suggests that a middle-out approach should be adopted. This approach identifies a similar gap, in that there is no stated direction to which way to go next (whether up or down).

Booch (1991), Coad and Yourdon (1991), Wainwright (1996), Avison and Fitzgerald (1995) and Hoche (1996) have all cited work on Object Oriented Analysis (OOA). Their work collectively suggests that OOA is a method of representing information in a distinct way. This means that part of the problem is identified and recorded showing how the attributes within the whole problem are connected with each other. This method requires detailed knowledge of the requirements of the system, suggesting that the developer should interact with the user to identify the information needed to develop the system. Structured Brainstorming was reviewed by Byrne and Barlow (1993). Their work suggests that Structured Brainstorming should assist the user in thinking further to identify the proposed system.

Mumford's (1994) Effective Technical Human Implementation of Computer Based Systems (ETHICS) method, which has also been evaluated by Wong (1994), argues that the social aspects of systems development are important and Mumford (1994) pays attention to this within ETHICS.

The methodologies have been reviewed and the question 'are the methodologies that have been reviewed considering user involvement?' has been answered. The answer is 'yes'. However, how should the interaction between the developer and the user take place and how should the requirements be captured? The answer according to the current research relating to the methodologies reviewed in this chapter is not clear.

Methods such as CIMOSA have a requirements definition and design stage. SSADM has a requirements specification and analysis stage. IDEF0 suggests that the As-Is and the Should-Be model should be built. RADD, RAD and JAD uses workshops to involve users. But 'how do we complete these tasks?' 'How do we specify the requirements?' Most of all 'how do we capture the information from the users?'

The methods do not appear to identify who should be involved and how they should be involved. Davis et al (1985) and Leonard-Barton (1988) state that less experienced users prefer to be elected rather than a representative being selected, while the opposite is true for more experienced users. However, the election procedure may take time that may not be available. It is likely that a representative needs to be selected to act as the bridge between the users and the developers. However, Carnall (1986) and Boehm and Ross (1989) state that mutual trust between users from various departments of the same organisation often does not exist, and cannot be established overnight. In addition, working together often brings conflicts of interest to the surface, which have to be dealt with explicitly. This suggests that communication skills within the design group have to be learnt, since users with little technical expertise cannot understand the jargon used by expert designers. Carnall (1986) also states that the communication element within systems development needs to be addressed. This is just one example of how complicated the process can become. Therefore, simplicity of the method becomes very important as the process does not want to be so complicated that the users need extensive training before they can understand the method.

Another consideration that was stated by Boehm and Ross (1989) is that users prefer the process to be carried out in groups, since they can help each other to remember their jobs by prompting each other. It also brings different parts of the organisation together. This may help the various departments gain an appreciation of what the other departments are doing.

Barki and Hartwick (1989) state that there have been suggestions to define and measure user participation as a set of operations or activities performed by individuals, or as a subjective psychological state. However, there seems to be little in the way of a formal process which describes user participation from its initiation, follows it through and measures any results or consequences of feedback.

Authors, including Toh et al (1997), argue that a major part of the IS design process is the specification of the information requirements. Zwegers et al (1997) state that active human involvement is important for successfully dealing with unforeseen events and complex situations in daily operations. Therefore, this discussion concludes that user involvement is an important part of systems development and the methodologies acknowledge this in their phases, stages or descriptions. However, they could be improved by clearly specifying how to involve the users from the beginning to the end of the requirements capturing process.

Table 2.4: Identification of the strengths and weaknesses of systems development methodologies in relation to capturing user requirements (adapted from Ince and Kay, 1998)

METHOD	DESCRIPTION	STRENGTHS	WEAKNESSES
SSADM Structured Systems Analysis & Design Methodology (Weaver, 1993) (Longworth, 1992) (Ince, 1996)	Uses various diagrammatic techniques, to show the processes that are relevant to the required system.	Uses detailed rules and standards to help structure the development process.	The method is rigid and does not show how the user and the developer should interact.
YSM Yourdon Systems Method (Yourdon, 1993) (Avison et al, 1995)	Used to break down a system into easily definable components.	Very good for defining the present system and is not complicated. Not very rigid.	No specified method for involving the user. Very brief structure. Needs more controls.
ETHICS Effective Technical and Human Implementation of Computer-Based Systems (Mumford, 1994) (Mumford, 1979) (Wong, 1979) (Avison et al, 1995)	Defines the social perspective on involving users in systems development.	Design separates social and technical perspectives and then brings them together later.	The interactive activities are not clearly defined.
DSDM Dynamic Systems Development Methodology (Stapleton, 1997) (Carter, 1997)	Uses categories of well-defined prototypes for use within the development life cycle.	Well structured method that encourages user participation and prototyping for better results.	No exact structure for capturing user requirements. However, guidelines are used for involving the user.
RADD Rapid Application Data Development **RAD** Rapid Application Development **JAD** Joint Application Development (Roche, 1998) (Martin, 1991)	Uses group sessions to extract information from users. Group sessions are used. Encourages speed in the analysis process. Groups session and CASE tools are used to represent the processes.	Brings users together. Interactive process, which uses software to represent results immediately. Very fast method of development. Good approach to getting the user involved. Brings the user and the developer together	Hard for participants to understand. Need for extra training which strains resources such as time and money. Harder to get very specific information. Better used as a brainstorming session. Problems could come about due to technology overpowering the whole process.

METHOD	DESCRIPTION	STRENGTHS	WEAKNESSES
OOA Object Oriented Analysis (Booch, 1991) (Wainwright et al, 1996) (Coad and Yourdon, 1991) (Hoche et al, 1996)	Defines areas of systems in a way that information is related to a specified area.	Good for identifying where information originates from.	No major user strategy. It aims to give developers a method for use within their own techniques.
SB Structured Brainstorming (Byrne and Barlow, 1993)	Method used to identify changes in present system.	Helps user to think further than normal.	No structure for capturing requirements.
GRAI Graphical Results and Activities Interrelated Method (McCarthy et al, 1994) (Wu, 1994) (Domeingts, 1985) (Chen et al, 1997) (Komi, 1995)	Tool used for the analysis and design of planning and control systems	Uses a structured grid to identify the importance of information. Good for analysis purposes.	Users are not seen as a major factor. The processes are shown as very broad areas. Very complicated method.
IDEF$_0$ ICAM Definition Method (Busby et al, 1993) (Cantemessa, 1998) (Wu, 1994) (Komi, 1995)	Uses boxes and arrows to represent activities, inputs, outputs and controls.	Very good for modelling processes. Has set boundaries on diagram sizes.	Lack of specific user input. However, is very useful for representing basic process information.
GERAM Generalised Enterprise Reference Architecture and Methodology (Vernadat, 1996) (Bernus and Nemes, 1996)	Identifies sets of components necessary for integration of enterprise architectures.	Life cycle approach shows a specific beginning and end to the definition process.	No real user consideration. Also lacks a prototyping stage, which is necessary for testing purposes.
CIMOSA Computer Integrated Manufacturing Open Systems Architecture (Kosanke et al, 1997) (Kosanke, 1995) (Toh et al, 1997)	Architecture used to streamline systems to the needs of the enterprise.	Separates the system into various sections. This helps to produce building blocks which can be worked on individually.	No user participation identified. It is also very complicated and requires a great deal of expertise in systems development.

Table 2.5: Comparison of the selected methods identifying whether they consider various factors relating to user involvement

Method	UI	U/DIP	RS	RA	SD
SSADM Structured Systems Analysis & Design Methodology (Weaver, 1993) (Longworth, 1992) (Ince, 1996)	Possibly	Not clear	Clearly	Clearly	Clearly
YSM Yourdon Systems Method (Yourdon, 1993) (Avison et al, 1995)	Clearly	Not clear	Possibly	Not clear	Not clear
ETHICS Effective Technical and Human Implementation of Computer-Based Systems (Mumford, 1994) (Mumford, 1979) (Wong, 1979) (Avison et al, 1995)	Clearly	Not clear	Clearly	Possibly	Not clear
DSDM Dynamic Systems Development Methodology (Stapleton, 1997) (Carter, 1997)	Clearly	Possibly	Clearly	Clearly	Clearly

Method	UI	U/DIP	RS	RA	SD
RADD Rapid Application Data Development **RAD** Rapid Application Development **JAD** Joint Application Development (Roche, 1998) (Martin, 1991)	Clearly	Clearly	Clearly	Not clear	Clearly
OOA Object Oriented Analysis (Booch, 1991) (Wainwright et al, 1996) (Coad and Yourdon, 1991) (Hoche et al, 1996)	Possibly	Not clear	Clearly	Clearly	Possibly
SB Structured Brainstorming (Byrne and Barlow, 1993)	Clearly	Not clear	Possibly	Possibly	Not clear
GRAI Graphical Results and Activities Interrelated Method (McCarthy et al, 1994) (Wu, 1994) (Domeingts, 1985) (Chen et al, 1997) (Komi, 1995)	Possibly	Not clear	Possibly	Clearly	Possibly
$IDEF_0$ ICAM Definition Method (Busby et al, 1993) (Cantemessa, 1998) (Wu, 1994) (Komi, 1995)	Clearly	Not clear	Clearly	Clearly	Possibly
GERAM Generalised Enterprise Reference Architecture and Methodology (Vernadat, 1996) (Bernus and Nemes, 1996)	Clearly	Not clear	Possibly	Possibly	Clearly
CIMOSA Computer Integrated Manufacturing Open Systems Architecture (Kosanke et al, 1997) (Kosanke, 1995) (Toh et al, 1997)	Possibly	Not clear	Clearly	Clearly	Clearly

UI – User Involvement, **U/DIP** – User / Developer Interactive Process,
RS – Requirements Specification, **RA** – Requirements Analysis, **SD** – Systems Design,

2.2 The problem

2.2.1 Why Information Systems succeed or fail

There is continuing difficulty in achieving success with IS, particularly in the sense of meeting user needs and expectations. This suggests that a fresh examination of the issues needed so that we can better understand the causes of success and failure (Whyte and Bytheway, 1995).

Poulymenakou (1996) states that IS failure is discussed predominantly in terms of monetary losses. Carnier (1958) states that success is measured with the automation of the whole office. Whyte and Bytheway (1995) state that experts working with information technology often perceive successful systems to be concerned with the successful use of the latest technology. This may not be the solution in todays workplace, since manual methods usually form some form of advantage. This is measured by evaluating various areas of the organisation and determining where IS are needed and where manual methods show more benefits. Project managers view success as a project that has been completed in the face of enormous difficulties (probably self-inflicted). These examples show that there is still a great deal of room for improvement within the attitudes toward success.

Ewusi-Mensah et al (1995) have suggested a few classifications of factors that determine the success or failure of an IS project. These are as follows:

(1) the type of decision or task the project was intended to accomplish
(2) the perceptions of senior management with respect to the risks / complexity and benefits the project was anticipated to experience.
(3) the strategic significance of the project to the organisation.

Point (3) carries the most importance, since the aim of any activity performed by the company should eventually show a significant strategic benefit to the basic well being of the organisation. If, as suggested in point (1), the project does not fulfil its intended objective, but benefits the company in another way, then maybe the project should not be deemed as a failure, but rather, a success. This should be the aim of any manager, especially if new technology is introduced while a project is ongoing. This really means that wasting resources has no benefit. The project must be diverted and used to solve a different problem whenever possible.

When examining the issue of user involvement in systems success or failure, it could be logical to mention the socio-technical factor. This has been mentioned by Mumford et al (1979). They state that this socio-technical feature involves user participation in systems development, which is shown in the ETHICS methodology.

Cale and Curley (1987) have stated that user involvement leads to more successful systems. It appears that the type of system that is being produced leads to the degree of user involvement necessary. This does make sense, but the main determinant of the degree of user involvement should be the amount required for the user to be satisfied or to feel the need to use the system and the degree used to make the system create a benefit for the organisation. Lacity and Hirschheim (1993) state that systems which are of a more critical nature tend to require more user participation.

Literature produced over the last 10 years has a great deal of respect, or shows a great need for user involvement when determining the success of an IS. Amoako-Gyampah and White (1993) have suggested that it is the value which is attached to this involvement by the development teams that appear critical, not the participation itself. Norman (1988) states that all software development should be considered as the combined results of human and organisational elements. He also states a few system images that reflect quality within a system, which would be a measure of the system's success. These include:

(1) The principles of system operation need to be observed.

(2) All systems actions needed to be consistent with the conceptual model (design model) of the system.

(3) The conceptual model that is developed for the system should be appropriate for the particular user, should capture the important parts of the operation of the device, and should also be understandable by the user.

These factors suggest an emphasis on ensuring a usable system that is recognised by many levels of the organisation, which includes management, end-users and developers. Therefore, the results of the use of the system will be seen clearly by the performance of the organisation.

Factors that determine systems success are rare. This may be because the benefits are usually expected, and when the success occurs it is usually shown in the form of growth. Rivard and Huff (1988) have found the following dimensions to be good predictors of IS success:

(1) degree of user independence from the systems department
(2) satisfaction of the set-up of the information system
(3) user friendliness
(4) user attitudes
(5) satisfaction with the degree of support received from the information system

Magal (1991) has also identified three measures

(1) quality of service
(2) quality of user-developed applications
(3) degree of user self-sufficiency

It appears that the user plays a large role in the success of a system, but there does not seem to be that direction towards implementing the ideas into systems development methods. However, there has been an increase, but systems are obviously in need of more attention in the development / design stages. This means that something must be done to decrease the high level of failure in the development of IS. A few of the methods such as Return on Investment (ROI), Cost / Revenue analysis and Cost Benefit Analysis have been studied by Ballantine et al (1993).

Financial analysis provides methods for measuring success. These measures show reasonably good estimations of how financially beneficial an IS implementation has been. One of the most commonly used methods is Cost Benefit Analysis (CBA) (Ballantine et al, 1993). This method requires costs to be compared directly with benefits that occur as a result of a particular investment.

The methods mentioned above calculate figures that relate to the investment and show success or failure as a monetary form only. Within the equation, there should be something that accounts for the intangible factors, which must affect the overall success of the project.

Ballantine et al (1993) state that many of these criteria are being widely used to assess and justify IS investments. Their study has formed a set of results to show the levels of usage of the techniques. These are as follows:

CBA was employed in 72% of cases
Payback Period in 60% of cases
ROI in 43% of cases
NPV in 25% of cases
IRR in 24% of cases

They also found that more than one technique was employed by organisations.

Parker et al (1989) have developed a technique called Information Economics. This is a particular variant of a set of techniques, which are usually referred to as multi-criteria techniques. These techniques are often regarded as alternatives to cost benefit analysis. The principle is based on the view that there are measures of worth other than monetary measures. The method appraises the relative usefulness of different desired outcomes and attempts to rank them by applying a preference weight to each goal.

Factors related to the overall success of the system usually relate to how the system is provided by the developing teams, and how it is produced in relation to user needs and expectations. The developer should try to identify the parties and attributes that need to be considered. Then the attributes or factors should be satisfied, therefore there is a higher chance that a usable and beneficial system will be produced.

2.2.2 The importance of user involvement

The reasons for the failure of IS are not well understood. Research is needed to understand why so many systems fail and to identify and / or develop tools and techniques to aid the successful development and implementation of IS (Tait and Vessey, 1988).

Cheney and Dickson (1982) state that, according to their research findings, the end user has become crucial to the success or failure of a system. Benjamin (1982) conducted a study and reported various results including, in 1981 25% of Xerox Corporation resources were dedicated to computer end users. By 1991, he states that the percentage is expected to triple. This type of evidence leads to various assumptions. The first being that the end user is a very important element and also that methods to make the job of interacting with the user easier is vital to the success of a system.

When the end user is considered, the company is working towards improving various elements that effect how productive a worker can be. For example, Yaverbaum (1982) states that the computer affects worker satisfaction, worker motivation and job content. These can eventually have an effect on the use of information and the resources of the company. Yaverbaum (1982) also states that if the organisation hopes to realise the optimum productivity promised by high technology, there is an urgent need to focus attention on the user.

The design element of a new information system is the major area of concern within this study. There will eventually be an optimum result to improve or begin to start the improvement process of systems design. This statement is in accordance with Cheney and Dickson's (1982) thoughts on the assumption that a satisfied end user will be a productive employee and that a major objective in the design and use of IS should be the satisfaction of the end user.

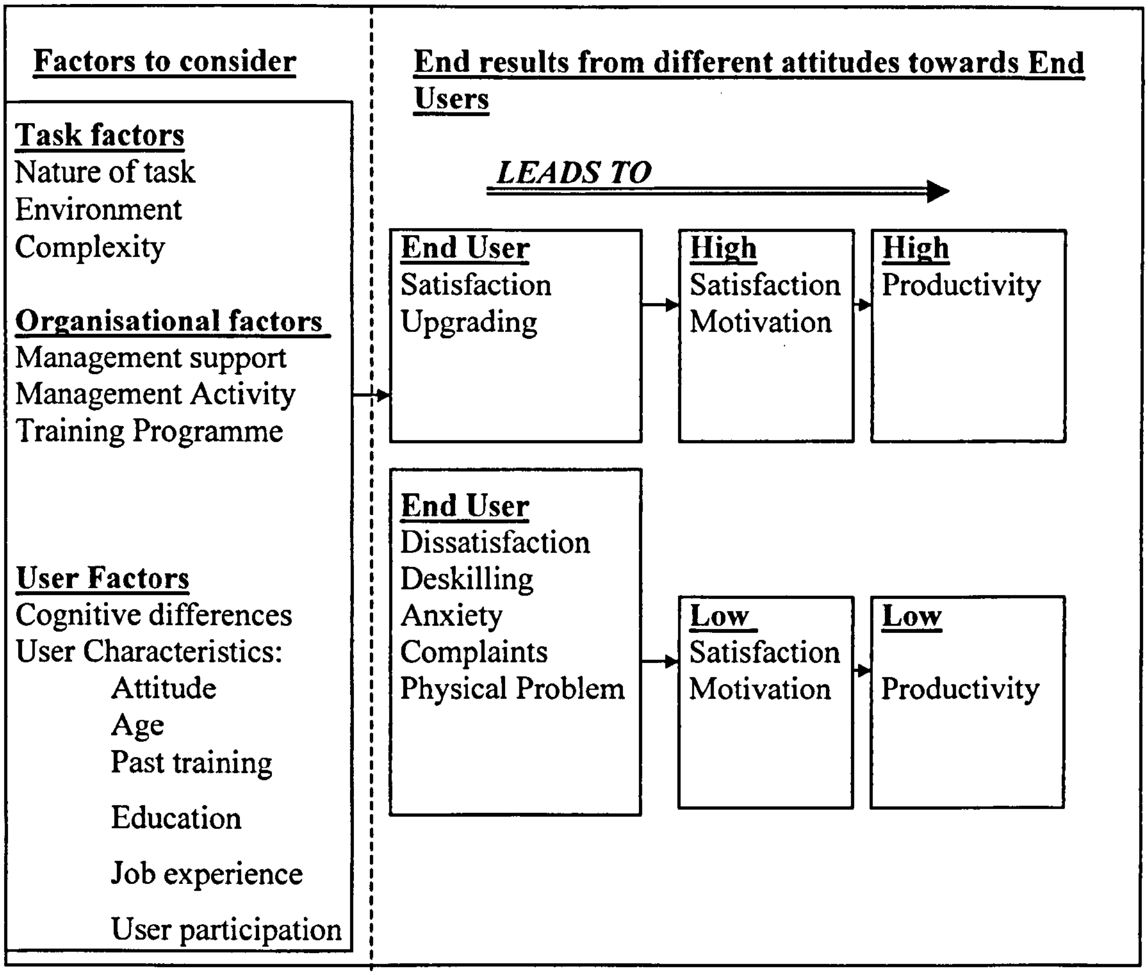
Howard and Smith (1986) and Leonard-Barton (1985) have created a short list of goals that would lead to an increase in the involvement for the end users, which should eventually increase worker satisfaction. These are as follows:

(1) Increasing user participation in the design of a system
(2) Understanding user characteristics that foster system acceptance
(3) Understanding organisational and task factors that affect user needs.

Figure 2.7 provides an understanding of how various factors affect the performance of end users. The diagram shows whether the user, task or organisational factors lead to end users feeling satisfaction or dissatisfaction. The level of satisfaction will result in the level of motivation felt by the employee, which will eventually lead to the effect the system has on productivity.

The financial elements associated with systems development are also very important (Ballantine et al, 1993). In order to keep costs to a minimum, the systems development projects should be evaluated according to the time that it should take for the project to be completed. Therefore, it is likely that the development process could overextend the time barriers. McFarlan and McKenney (1983) state that certain implementations, even those in the 1980s, run well over budget. Some are discontinued, while others perform at levels far below those expected. They also state that this problem is still quite common today. Todays problem involves companies seeking the "quick fix" option where software is purchased and it does not cater for the companies longer term needs.

Figure 2.7: Factors affecting End Users (Yaverbaum, 1982)



Alter et al (1978) has reported that a positive relationships has been found between user involvement and systems success.

Figure 2.8 presents the model that has been tested in a study by Tait and Vessey (1988). This diagram illustrates two propositions that were tested in their research:

Proposition 1: The extent of user involvement in successful systems design and implementation is affected by user system, technical system, and development process variables.

Proposition 2: The success of a system is affected by the extent of user involvement in its design and implementation, user system, technical system, and development process variables.

Figure 2.8: Contingency Model of user involvement and successful system design and implementation (Tait and Vessey, 1988)

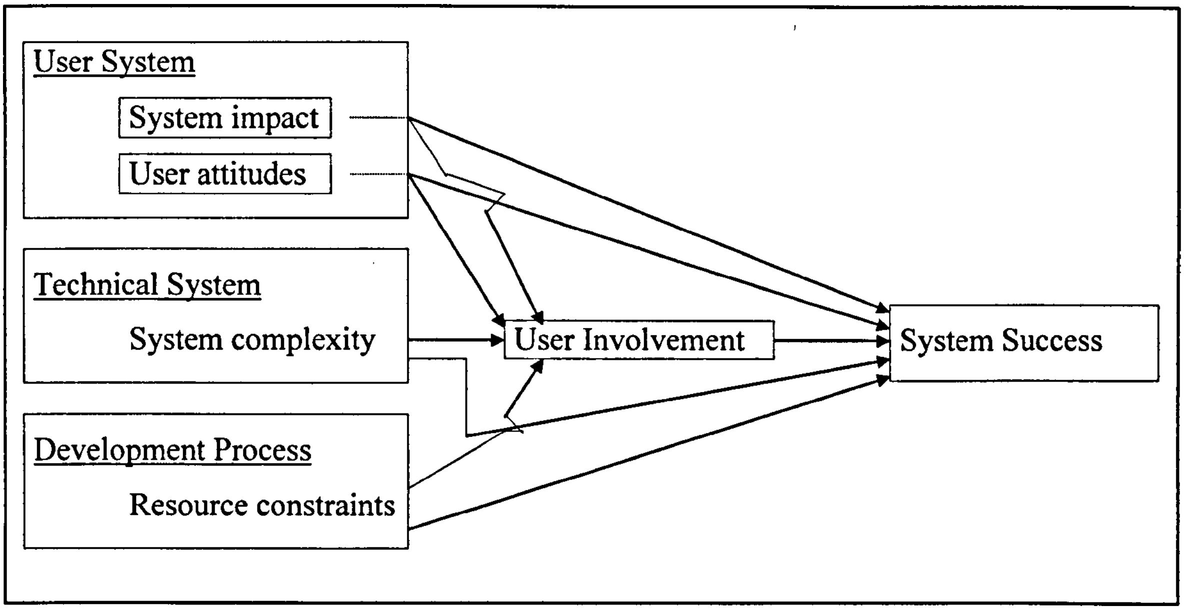
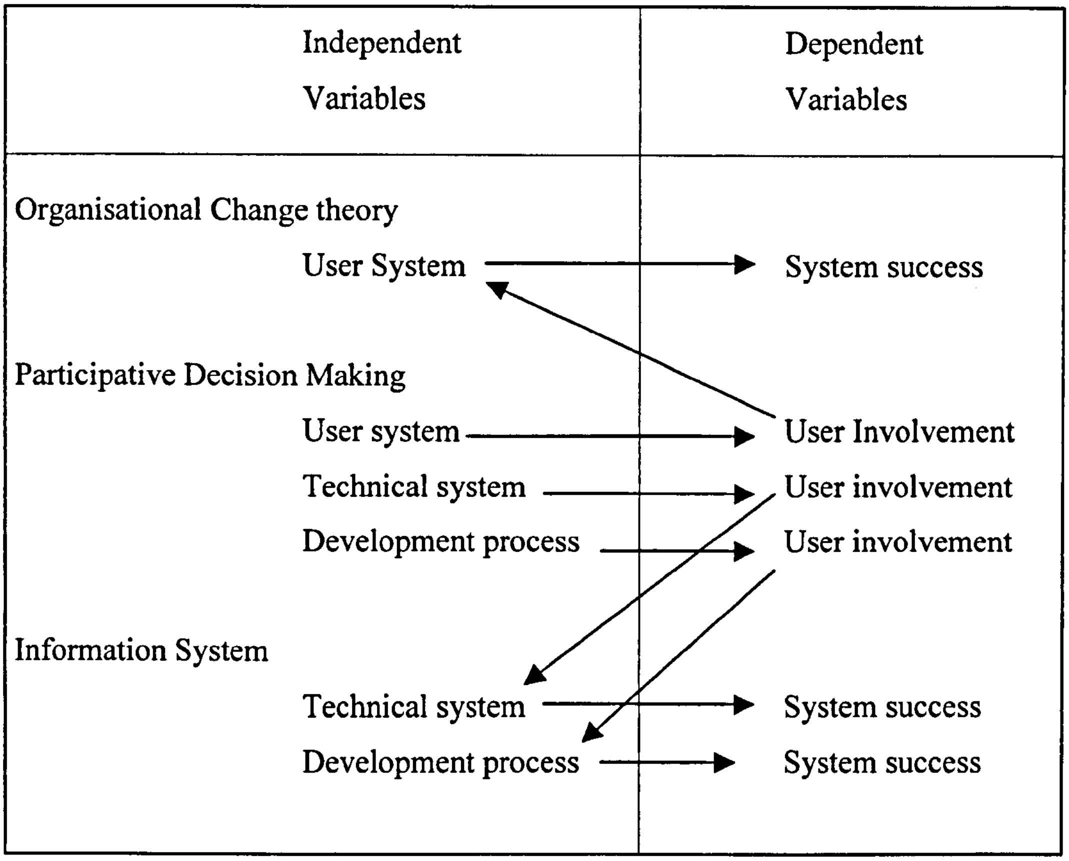


Figure 2.9 represents the theoretical research results, which form the links between the types of contingent variables and the extent of user involvement and successful system development and implementation.

Tait and Vessey (1988) have attempted to identify how user involvement can affect the success of a system (see figures 2.8 and 2.9). Their research concludes that user involvement has a positive effect on systems success. However, their work suggests that the affect is not significant.

Figure 2.9: Theoretical links between types of contingent variables and user involvement and systems success (adapted from Tait and Vessey, 1988)



Olson and Ives (1981) defined user involvement as the participation in the development by a member or members of the target user group. For example, user involvement could relate to the user at each stage of development including design, development, implementation and usage.

The extent of user involvement has also been categorised by Ives and Olson (1984). This list is formed in the following way :

(1) No involvement

Users are unwilling or not invited to participate

(2) Symbolic involvement

User input is requested but ignored

(3) Involvement by advice

Advice is solicited through interviews and questionnaires

(4) Involvement by weak control

Users have "sign off" responsibility at each stage of the systems development process

(5) Involvement by doing

A user is a design team member, or is the official liaison with the information systems development group

(6) Involvement by strong control

Users may pay directly for new developments out of their own budgets, or the users overall organisational performance evaluation depends on the outcome of the development effort.

From the work discussed above, there appears to be few negative effects when involving users in the design process. However, Keen et al (1987) state that it can be dysfunctional, in that it can lead to political problems. These views were gathered in the 1970s, when it was uncommon to involve users in the development process. At that time, it was more common for this work to be placed in the hands of the development team. Users were also minimal together with the number of machines that were present in most organisations. In these cases the old fashioned method has a better chance of working without effecting the productivity of the organisation. Today, however, there is a much higher proportion of employees operating computer systems. This means they will need a system they can understand, and can also use to the company's benefit.

A method of development that involves users is participative design. Hirschheim (1985) states that this method not only broadens the base of user involvement, but also places the responsibility and control of the design process in the hands of the user. This idea is extreme and requires a great deal of acceptance and development knowledge. However, if this goal can possibly be achieved, there will be a high degree or productivity and benefits for the whole company.

Mumford (1983) has identified three types of involvement that are quite similar to Olson and Ives' (1981) categories. These are listed below:

(1) Consultative

Where the users are consulted about what they want but where the bulk of the decisions are left to some other group.

(2) Representative

Where a group of users are elected to represent the needs of their co-workers in the design process.

(3) Consensus

Where the users not only make the decisions, but assume full responsibility for their implementation.

Participative design which has been mentioned by Hirscheim (1985) is closer to Representative and Consensus, since these allow more input from the users giving them more responsibility and authority over the development process. Once again the appropriate type has not been specified. This could mean that companies should use the method that is most appropriate for their working practices and the types of employees.

Lawrence (1993) states that user led development is gaining popularity with organisations wishing to increase user involvement. He has adopted this idea from representative user involvement. In this case, the user representatives will define the system requirements, oversee systems testing, manage the training and systems development and manage the overall project. Control is shared by the user and the systems personnel in another form of representative user involvement. This idea is user led and requires close contact with the user. This will eventually mean that methods would be required to make this task more efficient. If this can be achieved then high levels of satisfaction should be gained by the company. This is due mainly to users being a part of the employee group. Therefore, it is likely that the social and psychological aspects will be taken into consideration.

2.3 User Involvement in Information Systems development

2.3.1 The evolution of user involvement

In the 1980s and early 1990s literature on end user computing really began to identify its relevance in corporate computing environments (Alavi, 1990). Benson (1983), EDP Analyzer (1983), Van Kirk (1995), Caginalp (1994), Burrows (1994), Igbaria et al (1989) and Henderson and Treacy (1984) have all identified that it is a rapidly expanding phenomenon and is of growing strategic importance to many corporations. However, Alavi and Weiss (1985) and Davis and Olson (1985) state that, despite the general enthusiasm for end-user computing activities, there is a growing concern over the organisational risks and costs of these activities. Alavi (1990) states that this is because end-users are not necessarily trained data processing professionals and some user-developed systems may be poorly designed, thereby leading to potentially fatal consequences for an organisation's activities. Therefore, how far do we go with this process of involving the user?

This method of involving the user presents a vast range of opportunities including End-User Computing. Bigler (1995) has discussed the idea of developing End-User computing guidelines. He states that more often, the end-user of all types of computing are developing their own applications without the aid and professional know-how of an organisation's centralised information services department. However, the systems that are developed in this way may run the risk of having inadequate controls, documentation and other elements that more experienced professionals would adopt from past experience.

Gilbert (1978) states that most managers saw computers as tools for providing information on which to base their decisions. Today, the research has examined this situation and studies show that the dependency on the systems has become dangerous, due to a lack of trust in what is being produced by the system. Pourmaleski and Cardinali (1994) state that people who are successful in computer integration in the business know that when developing computer applications they should first focus on their information needs, rather than on hardware technology. They realise that it is essential to know the information requirements of management, and how computers can be used to meet those requirements.

The area of capturing user requirements has been left to the individual who is responsible for the process. However, the exact structure or method to carry out this process is not easy to identify, leaving no structure for improving the process. In order to find a starting point we can look into the development stages in the IS development cycle, and examine and analyse the various constraints that deter us from reaching our desired goals. Within research papers by Corbett et al (1991) and Markus and Bjorn-Andersen (1987) there is an argument that the use of structured methodologies to support ISD leads to fragmented, highly specialised, low-discretion (i.e. de-skilled) jobs for systems users. Corbett et al (1991) argues that this approach to using structured methods causes extreme task fragmentation, since there is little consideration of user task identity or the user's mental model of the task performed.

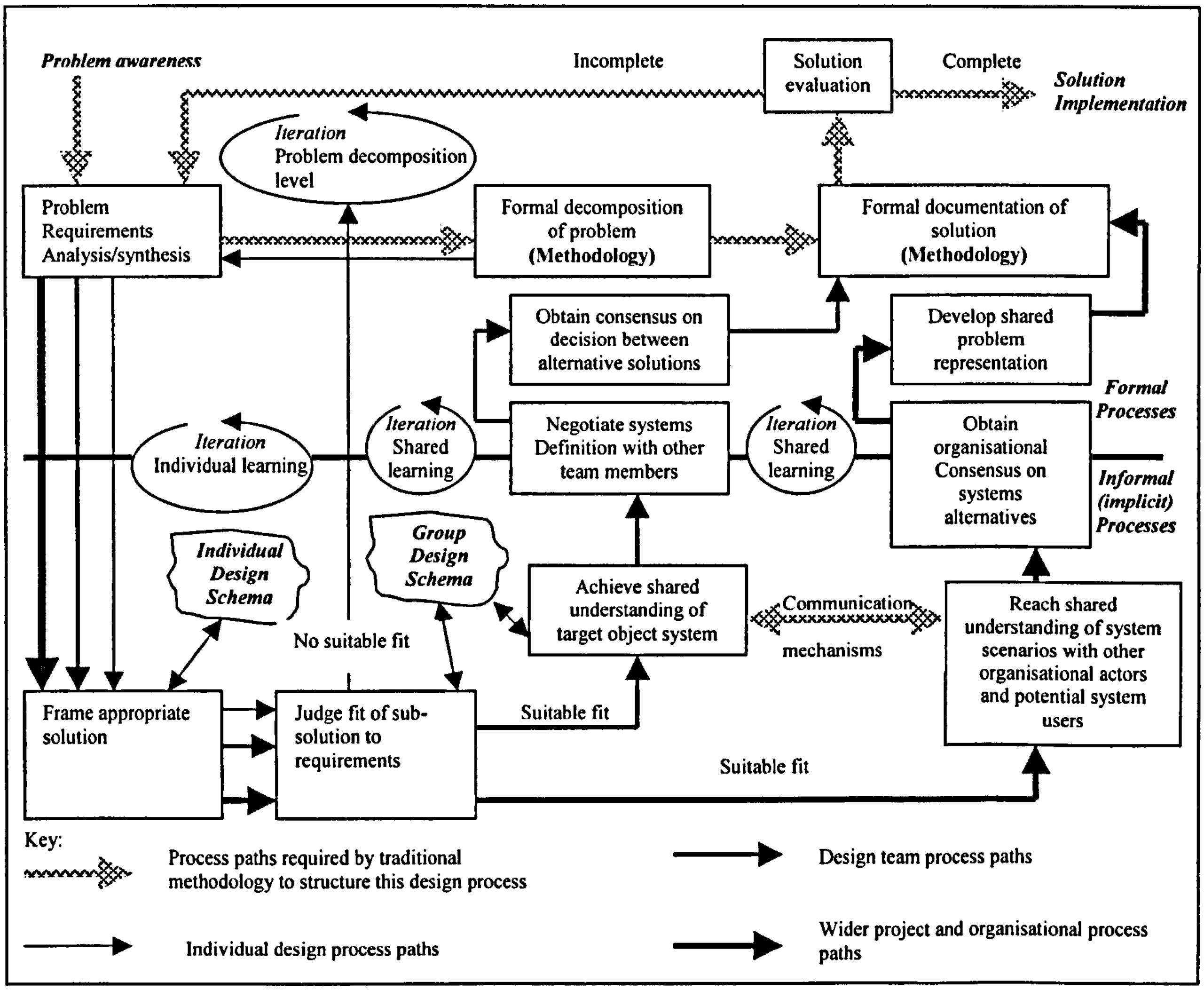
According to researchers such as Colter (1984), Couger et al (1982) and Gremillion and Pyburn (1983) many methodologies, techniques and tools exist and continue to be developed for accomplishing various tasks in the systems life cycle. Gasson (1993) states that a valid process model must take account of social interactions: the elements of communications and of learning, which takes place during the Information Systems Design (ISD) process. She has developed a model, which is shown in figure 2.10. She states that activities may take place on multiple levels at the same time. She also states that a comparison of the route through design activities is made with the route through the theoretical design activities presupposed by the waterfall model (see figure 2.2) underlying structured development methodologies.

Even though the model is not really promoting a structured approach, figure 2.10 offers a range of opportunities for this research as it brings together the ideas that help to develop and implement a solution. The problem requirements shows connections to the problem awareness, solution evaluation, formal decomposition of the problem (methodology) and the frame appropriate solution. It can be observed that Gasson

(1993) has paid attention to the identification of the problem, while ensuring that there is adequate management control of the aim of the system. It also identifies the different players as individuals and as groups, where the model encourages shared learning.

Edwards et al (1995) states that when the formal IS development process begins, outline processes and entity models are used as the basis for much more detailed business modelling with operational management and staff. However, they state that from these, data flow diagrams are developed to specify the detail of the business processes, and data models specifying the detailed content of the databases; this then constrains the content of printed reports and enquiry screens. They have also developed a model (the V model), which they state is the traditional model for systems development which is shown in figure 2.11 (here it is adapted to strategic projects). This model does not have as much detail as the one proposed by Gasson (1993). However, they both show the requirements stage as an important element of the model. Edwards et al (1995) argue that needs analysis is very important to the success of the project and must be supported by appropriate time and effort. They state that any short cuts taken at this stage might save a little time and money in the short term, but will incur orders of magnitude more cost before the project is successfully completed.

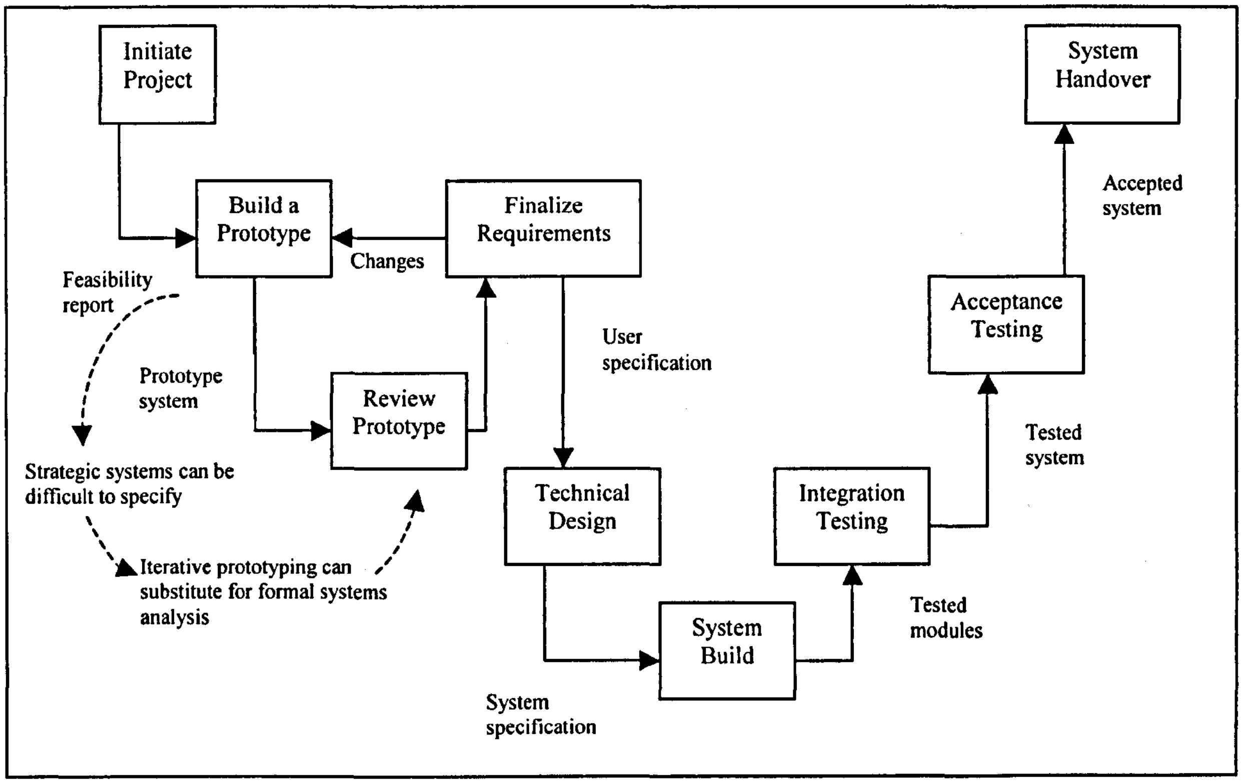
Figure 2.10: A proposed model of the design process (Gasson, 1993)



Taking into consideration these models and their attributes, it is possible to go deeper into the subject and identify how user involvement has been argued and discussed in the past. Nantz (1990) states that IS departments do not support end user applications, since they are not categorised by good design, programming, testing, and documenting techniques. She also states that additional problems affecting end users are the lack of good training programs and decreased worker productivity despite increased investments in hardware and software.

Tait and Vessey (1988) carried out a study and reported that, from their sample of 42 projects, increased user involvement reduces the risk of system failure in complex projects, Norton and Mcfarlan (1975) state that user involvement provides a more accurate and complete assessment of user information requirements.

Figure 2.11: Adapting the V model to strategic projects (Edwards et al, 1995)



2.3.2 Review of the methods designed to address user requirements

Various methods to assist the process of involving users have been identified and discussed. Table 2.6 shows the strengths and weaknesses that the methods presented.

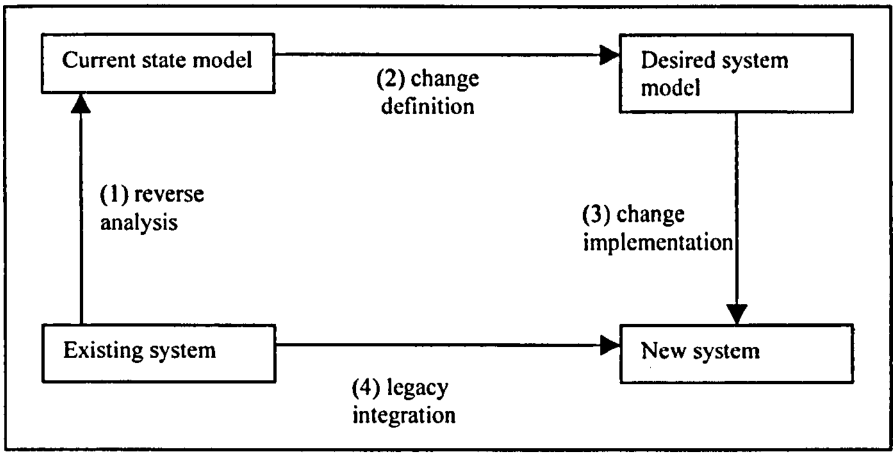
The traditional "waterfall" model of the system development process depicts of requirements gathering as a single step. Users, customers, and/or marketing representatives meet with designers, and describe what they want and /or react to what is proposed (Carroll et al, 1998). Reubenstein et al (1991) state that requirements acquisition is one of the most important and least supported parts of the software development process. Davis (1982) states that correct and complete information requirements are often difficult to obtain. He also states that there are three major reasons for the difficulty. These are:

1. The constraints on humans as information processors and problem solvers.
2. The variety and complexity of information requirements.
3. The complex patterns of interaction among users and analysts in defining requirements.

Brooks (1987) states that the hardest single part of building a software system is deciding what to build. No other part of the work so cripples the resulting system if done wrongly. No other part is more difficult to rectify later. Traditionally, the role of requirements engineering was to establish a complete, consistent, and unambiguous requirements that defines the requirements of the desired system at a conceptual level (Haumer et al, 1998). Haumer et al (1998) state that in the late seventies and during the '80s a variety of methods for defining the data (eg. ER modelling, Chen, 1976), the behaviour or the functions of the system (structured analysis) had been proposed. In the late 1980s and early 1990s, object oriented techniques appeared which proposed an integration of the three views.

Within the structure of systems development, the developer attempts to identify the As-Is (or Present) system requirements and then generate the Should-Be (Proposed) system. Figure 2.12 illustrates Haumer et al's (1998) 4 steps of a typical change process.

Figure 2.12: Four steps of a typical change process (Haumer et al, 1998)



The four steps are described as follows:

1) Reverse analysis: Defining a current-state model by abstracting from the reality is required since, in most cases, no conceptual model of the system exists or the model is not up to date;

2) Change definition: Integrating the change definition into the (partial) current-state model, thereby defining the model for the new system;

3) Change implementation: Designing, implementing, testing and installing the new system based on the desired-state model;

4) Legacy integration: Considering the existing context during the change implementation to empower reuse and to avoid conflicting system implementation.

Yu (1998) summarised some of the applications of goals in requirements engineering. He states that goals are used:

1. for requirements elicitation and elaboration, thereby encouraging the stakeholders to ask "why," "how else," questions.

2. for relating the system to organisational business context.

3. to clarify requirements and stakeholders objectives without the need to go into too much detail.

4. to deal with conflicts. Goals can be used as interconnecting mediation points supporting a focus on common objectives first, before going into the details of resolving the conflicts.

The four applications described by Yu (1998) identify the main uses of goals within this process. However, Potts (1997) states that it is unwise to apply goal based requirements methods in isolation and Rolland et al (1998) argue that, even though goal modelling is an effective approach to requirements engineering, it is known to present a number of difficulties in practice. They have carried out research examining goal modelling and scenario authoring to form a set of rules that form the basis of a software environment called L'Ecritoire, to guide the requirement elicitation process. The method is based on a requirements chunk model. This Requirements Chunk (RC) is the basic building block of the requirements chunk model, it is a pair (G, Sc) where the G is the goal and Sc is the scenario. A goal is defined by Plihon (1998) as something that some stakeholder hopes to achieve in the future, and Plihon (1998) defined a scenario as a possible behaviour limited to a set of purposeful interactions taking place among several agents.

The work of Reubenstein et al (1991) on requirements acquisition concludes that it is valuable for two reasons. First, from the perspective of software engineering, requirements acquisition is perhaps the most critical part of the software process. They state that studies indicate that errors in requirements are more costly than any other type of error. Second, from the perspective of artificial intelligence, requirements acquisition is a good domain in which to pursue fundamental questions related to knowledge acquisition in general.

Results from Rolland et al's (1998) evaluation show that elements were found to be useful in clarifying systems requirements, but the technical support was limited. Extensions and improvements such as supporting several natural languages, connection with industrial tools, providing co-operative negotiation and evaluation of requirements were suggested by workshop participants. Work carried out by the DSDM Consortium (1997) suggests that scenarios have been advocated as an effective means of communicating between users and stakeholders and anchoring requirements analysis in real world experience. However, Gough et al (1995) and Roger (1995) state that scenarios are extremely labour-intensive to capture and document. From a practical point of view, the use of scenarios can be time consuming.

Scenario-Based Requirements Engineering is intended to be integrated with object-oriented development. A separate requirements specification document is maintained to make requirements explicit and to capture the diversity of different types of requirements (Sutcliffe et al, 1998). This method is shown in figure 2.13. It is split into four stages, which are as follows:

1. Elicit and document use case. In this stage use cases are elicited directly from users as histories of real world system usage or are created as visions of future system usage. The use case model is validated for correctness with respect to its syntax and semantics.

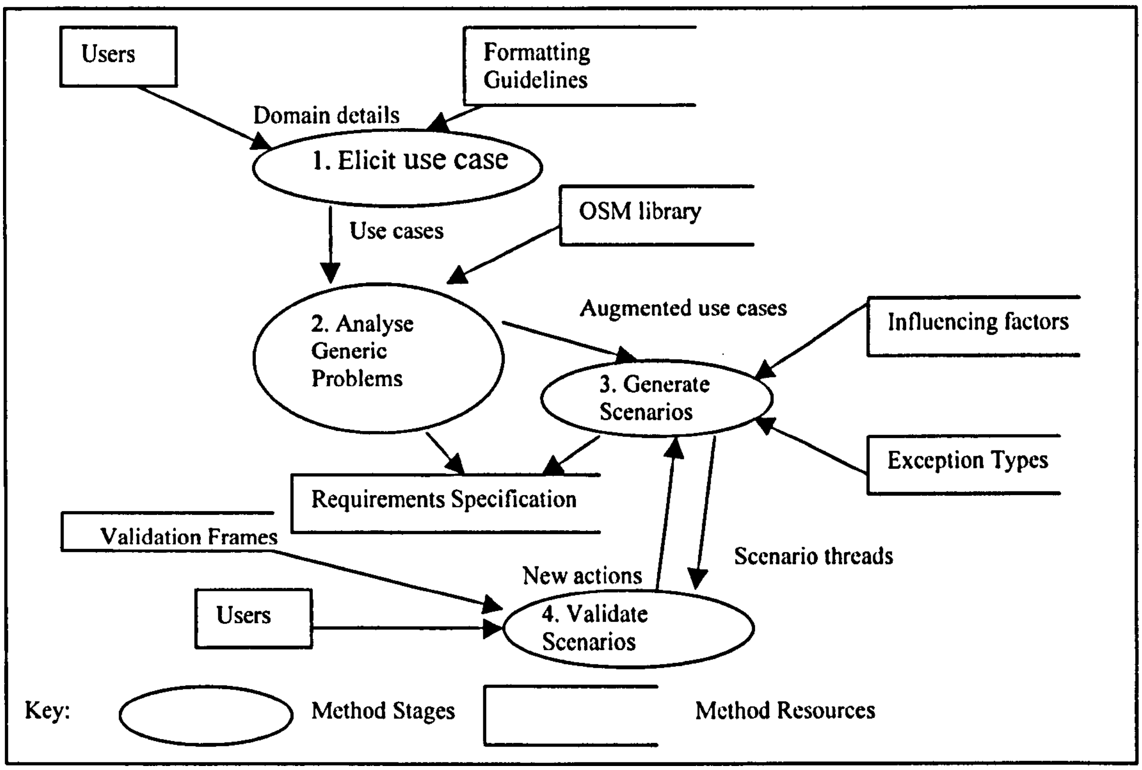
2. Analyse generic problems and requirements. A library of reusable, generic requirements attached to models of application classes is provided. A browsing tool matches the use case and inputs facts acquired from the designer to the appropriate generic application classes, and then suggests high level generic requirements attached to the classes as design rationale "trade offs".

3. Generic Scenarios. This step generates scenarios by walking through each possible event sequence in the use case, applying heuristics which suggest possible exceptions and errors that may occur at each step.

4. Validate system requirements using scenarios. Generation is followed by tool-assisted validation which detects event patterns in scenarios and presents checklists of generic requirements that are appropriate for particular normal and abnormal event patterns. In this manner requirements are refined by an interactive dialogue between the software engineer and the tool. The outcome is a set of formatted use cases, scenarios, and requirements specifications that have been elaborated with reusable requirements.

This approach was originally proposed by Jacobson et al (1992) and is based on the concept of scenarios. Lee et al (1998) state that it is arguably one of the best known and most widely employed requirements elicitation techniques in the industry. Software requirements are stated as a collection of use cases, each of which is written in the user's perspective and describes a specific flow of events in the system (Lee, 1998).

Figure 2.13: Method stages for Scenario Based Requirements Engineering (Sutcliffe et al, 1998)

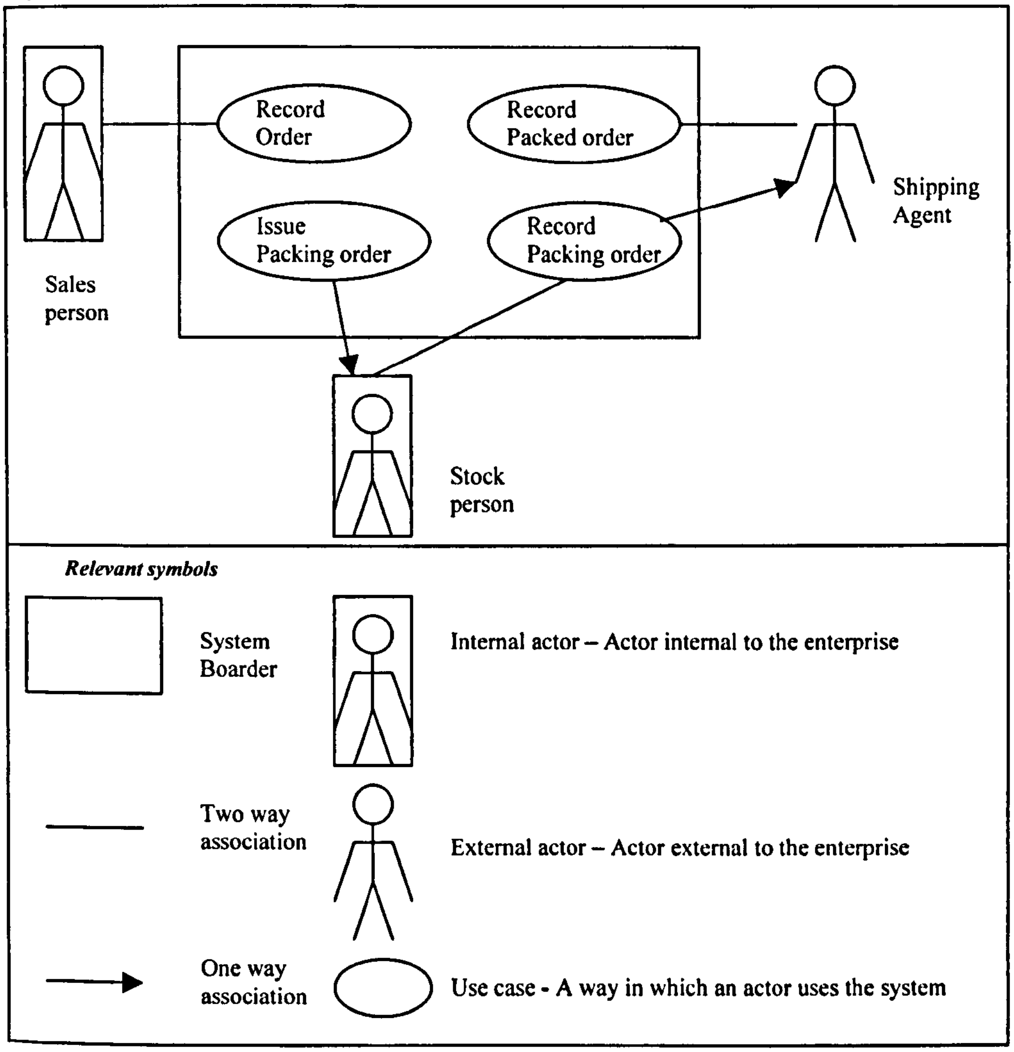


Allen and Frost (1998) state that the "non-technical" nature of use cases allows users to participate in a way that is seldom possible using the abstractions of object modelling alone. It also helps the analyst get to grips with specific user needs before analysing the internal mechanics of the system.

Allen and Frost (1998) have proposed a diagram to show how use cases can be used. This is shown in figure 2.14. The use case approach offers several practical advantages in that use case requirements are relatively easy to describe, understand and trace. However, Lee (1998) states that there are a couple of major drawbacks. Since use cases are often stated in natural languages, they lack formal syntax and semantics. It is also difficult to analyse their global system behaviour for completeness and consistency, partly because use cases describe only partial behaviours and because integration among them are rarely represented explicitly.

Reubenstein et al (1991) state that it is useful to distinguish three phases in the requirements acquisition process: elicitation, formalisation, and validation. The elicitation phase usually takes the form of a skull session, where the goal is to achieve a consensus among a group of users about what they want. In the elicitation phase, the requirements analyst acts primarily as a facilitator. The end product of the elicitation phase is typically an informal requirement. They state that most software requirements tools focus on the validation phase. The main goal of this part of the requirements acquisition process is increasing confidence that a given requirement corresponds to the end-user's desires. The focus of the Requirements Apprentice method is on the formalisation phase that bridges the gap between an informal and formal specification. This is a crucial weakness in this area of the research due to the simplicity that is necessary when interacting with less experienced users.

Figure 2.14: Use case diagram for ordering shipping system (Allen and Frost, 1998)



The work of Reubenstein et al (1991) has discovered some interesting characteristics that are involved with verbal communication between speaker and hearer. The method must take this into account and understand why it is necessary to understand the user more. The list of features are as follows:

1. Abbreviation – Special words (jargon) are used. The hearer is assumed to have a larger amount of specific knowledge that explains the words.

2. Ambiguity – Statement can be interpreted in several different ways. The hearer has to disambiguate these statements based on the surrounding context.

3. Poor ordering – Statements are presented in the order they occur to the speaker, rather than an order that would be convenient for the hearer. The hearer needs to hold many questions in abeyance until later statements answer them.

4. Contradiction – Statements that are true in the main are liable to be contradictory in detail. This reflects the fact that the speaker has not thought things out completely.

5. Incompleteness – Aspects of the description are left out. The hearer has to fill in these gaps by using his/her own knowledge or asking questions.

6. Inaccuracy – For all kinds of reasons, it is inevitable that some statements will simply be wrong in the sense that, while consistent and complete, they do not correspond to what the speaker has in mind.

The ADISSA (Architectural Design of Information Systems based on Structural Analysis) method described by Shoval (1988) is a refinement of Structured Systems Analysis, utilising enhanced DFD techniques not only to analyse the system but also to design it. There are two major products generated by the ADISSA method. One is the user interface, a menu tree, viewed as the external architecture of the system. The other is the internal architecture, a set of system transactions activated in response to various events and user requests. Babin et al (1991) argue that requirements are expressed in the ADISSA notation, using the ADISSA method, and provides consistency between specification and design. They state that elementary functions are incorrectly specified in many ways. The most common case is incompleteness.

The PAISLey method as cited by Zave (1991), is an executable specification language, accompanied by specification methods, analysis techniques and software tools. This method was the subject of a long-term research project from 1979 to 1987. She states that each specification language is intended to serve as the foundation for a streamline software development process in which only one complete formal description of the system is written.

Zave's (1991) work was carried out over a long period of time and she developed a language, but how do we put this in the hands of the user to get them more involved? It is possible that the placing information in the system could be carried out in a more structured manner but how do we know that the right information is being used?

A study carried out by Yadav (1981) identified some of the classical approaches to systems requirements analysis. These methods were discussed and conclusions generated that present an overview of how this area of the systems development life cycle was analysed. The main problems identified from this study's goals were in the user domain. Study Organisation and Plan (SOP) (IBM, 1961) identifies the flow of activities and describes the operations within these activities. However, Yadav (1981) states that it does not help to identify the need of the system. This point is very important for analysis purposes. The method to solve the problem of improving user involvement must consider such a factor as the final aim of the method should be to improve the effectiveness of the information received. Accurately Defined System (ADS) (Lynch, 1969) is very similar to Langefors Approach (Langefors, 1963). Both start with the assumption that the development starts after the user has defined the system. This returns to the study of end-user computing where it is necessary to understand what the user is capable of producing. The question here is how much responsibility do we give the user in the requirements capturing process? Yadav (1981) identifies them as methods that can be converted directly into machine language program by a compiler. McDonough's Approach (McDonough, 1969) recognises the linkage between organisational factors and IS factors, but does not provide any method as to how to identify and determine relevant factors. A useful method to the study lies in Young and Kents Approach (Young and Kent, 1958) which describes a method for formalising the documentation and data processing requirements area. They use four basic points, which Young and Kent (1958) suggest should be considered. These are as follows:

1. Information sets
2. Documents (inputs/outputs)
3. Relationships

4. Operational requirements

However, Yadav (1981) states that they do not pay enough attention to determining requirements. This is true but foundations are laid for the development of an individual company's method. However, this does depend on an informed study being carried out that identifies the needs of the method.

One of the main downfalls of the different approaches in relation to this study is that they lack the user consideration in terms of getting down to the users level as is shown in the use case method. There is good structure but they do not offer the requirements capturing stage sufficient consideration.

Lor and Berry (1991) state that systems design is a creative activity that requires a lot of human judgement. However, they argue that it is possible for the machine to assist in the process. An interactive tool is built to assist, but not replace, the human designer in building a system based on the requirements. The method is called Systems Architect's Apprentice (SARA). This is an environment-supported method for designing hardware and software systems. The goal of the tool is to ease the task of systems design within the SARA method.

Lor and Berry (1991) identified the importance of getting the requirements and the design on paper. Doing so forces more careful consideration and permits review by others with the aim of elimination of errors and verification that what is desired is expressed. They state that there are many language and textual or graphical techniques, for stating requirements. Some of the languages they mention include the Problem Statement language (PSL) by Teichroew and Hershey (1977), Requirements Statements Language (RSL) by Alford (1977) and Systems Verification Language (SVL) by Fischer and Walker (1979). They state that while some requirements handling tools assist in the preparation of requirements, none of them address the problem of producing a design from the requirements. The front end of the SARA method offers the use of Data Flow Diagrams (DFD) (DeMarco, 1979). However, after reviewing this idea based on their paper it can be concluded that the method does not step as far back in the process as would be expected and is too complicated for the users on the shop floor. However, there may be room for usage at higher levels.

A closer step towards the goal of the study is suggested by Kyng (1995). The method is participatory design. Kyng (1995) states that writing on participatory design often conflates requirements gathering with other system design activities, but much of the interaction among users and designers is typically directed at identifying and clarifying objectives of the design work. This view is similar to that of Paiget and Inhelder (1969) and Vygotsky (1978), who argue that requirements emerge through design activities: different design activities evoke different views of the requirements, and different requirements. However, the attempt to try different design methods often takes time and therefore may not be feasible in many cases.

The research offered by Davis (1982) also bears some significance to the requirements of the study. He has identified four strategies for carrying out this process. These are: (1) asking, (2) deriving from existing information system, (3) synthesis from characteristics of the utilising system, and (4) discovering from experimentation with an evolving information system. The work carried out by Davis suggests that the process is still in the hands of the developer, but more consideration is lent to the various factors that attempt to obtain information from the user especially that of asking, which has been broken down considering aspects such as open and closed questions, brainstorming, guided brainstorming and group consensus.

By reviewing some of the ideas presented so far, it is clear that the use case method offers the most feasible benefits to the study. The main reason for this is that it brings the requirements back down to the users view and diagrammatically represents the systems in a manner that could be understood by a less experienced IT user. However, as with most of the methods, they do not give the developer a framework for collecting the information from the user.

Table 2.6: Identification of the strengths and the weaknesses of the methods that have been designed to address user requirements

Method	Strengths	Weaknesses
Goal modelling & Scenario authoring (Yu, 1998) (Rolland, 1998) (Potts, 1997) (Plihon, 1998) (DSDM consortium, 1997)	Encourages users to ask "Who" and "How Else" questions. Greater benefits after requirements have been captured.	Does not specify how requirements should be captured.
L'Ecritoire (Rolland, 1998) (Potts, 1997)	Based on obtaining blocks of requirements, then creating goals and scenarios that are dealt with later. Goal – Something a stakeholder hopes to achieve Scenario – Behaviour limited to a set of purposeful interactions.	Lacks interactive steps so that users and developers increase their levels of interaction. This is used after the capturing process is carried out.
Scenario Based Requirements Engineering (Use-Case) (Gough, 1995) (Roger, 1995) (Sutcliffe et al 1998) (Jacobson, 1992) (Lee, 1998)	Uses low level requirements capturing process so that scenarios can be walked through. User interaction is paramount in this method.	Labour intensive and time consuming. Lacks formal structure. Difficult to analyse the global system.
Requirements Apprentice (Reubenstein, 1991)	Focuses the formalisation phase that bridges the gap between an informal and formal specification. Identifies what to consider.	Does not offer a guidance that takes the user through the whole process of user involvement.
ADISSA (Shoral, 1998) (Babin, 1991)	This is a refinement of structured analysis. It uses DFDs to analyse and design systems.	Can produce incorrect specifications due to its complexity.
PAISLey (Zave, 1991)	This is an executable programming language. Used for later stages once requirements are captured.	Not suitable for inexperienced users.
SOP (IBM, 1961)	Used for identifying flows of activities.	Does not help to identify the needs of the system.
ADS (Lynch, 1969)	Useful for the analysis of the requirements.	Used for analysis rather than capturing requirements.
McDonough's Approach (McDonough, 1969)	Describes the linkages between organisational factors of the system.	Does not help to highlight relevant factors that need further investigation.
Young and Kents approach (Young and Kent, 1958)	Formalises the documentation process of the end result of the user requirements capturing process.	Does not pay enough attention to determining requirements.

Method		Strengths	Weaknesses
SARA (Lor and Berry, 1991)		Tool used to assist the human designer. It recognises the importance of getting requirements on paper, which encourages careful consideration.	There is no defined structure for carrying out the process.
Languages	PSL (Teichrowd and Hershey, 1977)	Languages used for representing data procedures for high level design.	Lacks ability to interact with users at inexperienced levels.
	RSL (Alsford, 1977)		
	SVL (Fischer and Walker, 1979)		

2.4 The need for a new approach

According to Grindley (1995), there is a cultural gap between the IT developer community and the IT user community. In his survey of IT directors, 47% stated that their main problem was the cultural gap that exists between IT and business professionals, and 56% believed that the cultural gap was losing or seriously delaying IT opportunities for their company to gain a competitive advantage. This statement has helped to form a more in-depth idea of the relationship that exists between developers and users, or the lack of a relationship that exists between them.

Somogyi et al (1987) state that the problem of the lack of a relationship between developers and users has existed since the early 1950s. One attempt to solve this problem came with the introduction of the Systems Development Life Cycle, where the phases involved were developed to add some control and management to the process (Davis and Olson, 1985). This offers a good starting point which should encourage the proposed method to consider such a set of stages.

Since the late 60s various authors have attempted to find ways of identifying the user as an important asset to the development process. Mumford et al (1978), Land (1982) have examined design methodologies that identify more social aspects of systems development. There have been a number of ideas that show user involvement as an important factor, these include: Prototyping (Earl, 1978; Nauman and Jenkins, 1982;

Robey and Markus, 1984) and Third party intervention (DeBrander and Edstorm, 1977) These have all led to the bridge being constructed to help the user and the developer form a closer relationship.

This study intends to add to the research by developing a method of getting the user and the developer in close quarters for a better relationship that results in continuous feedback. Apart from these methods that look at the more social perspective, it is necessary that the system consists of accurate information. A survey of users and developers would form the results that show whether there is still a need for continued awareness and improvement of the user / developer relationship.

Fitzgerald (1996) states that after an examination of the literature, a two fold bias has evolved, which firstly construes the 'software crisis' (which is a word used for the problems inherent in systems development) as a problem arising from the sloppy, ad hoc and irrational approaches of systems developers in practice; and secondly, views the solution to the software crisis in terms of more widespread adoption of rigorous and formalised systems development methodologies. This statement shows that problems still exist in the methods used to develop systems.

Downs et al (1992) state that major institutions such as the UK government have mandated the use of the SSADM (Structured Analysis and Design Method) Methodology for systems development. SSADM is now used on projects totalling billions of pounds each year. Fitzgerald (1996) also states that this methodology has been adopted by several other national governments, whilst countries such as France, Holland and Italy have their own formalised development methodologies. This shows that there is an importance to formalising tasks, which could be adopted for the interaction with users. This is an area that could benefit from some formal structure. However, Humphrey et al (1991) state that effective development requires that all steps in a development methodology should be carried out regardless of circumstances. This statement makes it difficult to use a methodology, since every project is different and may have different needs with regards to structure. It may be sensible to attempt to keep to the specific steps to help the process to be carried out effectively, unless a new methodology is developed that takes into account the relevant steps from another method. This means that the process has a well defined beginning and ending. Fitzgerald (1996) states that the idea that the whole methodology needs to be used regardless fails to take into account the contingencies of any particular situation. Plavia and Nosek (1993) state that one startling and somewhat disturbing observation is that many (system development) methods are used very little.

Page-Jones (1991) states that the failure of practitioners to use a development methodology is seen as a weakness on their part. The trend seems to been to see how to get the user involved more in the systems development process, which is proven by various authors investigating the social side of systems development such as Mumford et al (1978), Earl et al (1988), Gibson and Nolan (1974). Today, however, the user has been getting closer to the production of systems especially in the form of End-User computing (EUC). Taylor et al (1998) states that End-User computing is a growing area within the Information Technology (IT) industry. The number and size of end-user development applications is steadily increasing, yet little attention is paid to the way in which such applications are developed and their impact upon the organisation. Edberge and Bowman (1996) state that the proliferation of personal computers within the organisation has meant that an increasing number of end users are now capable of potentially developing their own applications. This means that the users ability to get involved has definitely been increased.

Evans (1989) and Nantz (1990) state that end-users should develop and maintain their own systems to the same standards as the IT department. Research by Taylor et al (1998) indicates, that for the vast majority of end-user activities in the 34 organisations studied, the IS methodology used by the IT departments would be far too large and unwieldy. Table 2.7 shows some of the advantages and disadvantages of using an Information Systems Methodology (ISM) for end-user projects.

Mumford (1997) raised a question which was 'why isn't participative systems design more popular?' She states that it gives practical form to the philosophy of humanism by providing employees with an opportunity to influence the work systems that surround them. It also provides them with an improved work environment in which they are encouraged to increase their skills, make judgements and take decisions. She also states that one of the most pressing problems for todays managers is providing their companies with a degree of stability in times of rapid, continuous and multiple change. Rittel (1984) sees the design process as an argumentative process, which is not a well-designed sequence of activities that are planned and systematically carried out. It is argumentative because statements and solutions made and found are expressed (or made transparent) and are subject to the scrutiny of the participants. Different viewpoints are considered and debated. This process is iterative, solutions are generated from the arguments and these are further subjected to discussion and debated until most participants are satisfied or some sort of consensus is reached. Decisions evolve out of this social communicative and dialectic process among the participants and the designers. The design generated is not only more authentic to the needs of the participants, but will be more receptive and liked by the users involved.

Table 2.7: Summary of potential advantages and disadvantages of using IS methodologies for end-user computing projects (Taylor et al, 1998)

Advantages	**Disadvantages**
Reduce duplication	Unnecessary bureaucracy
Reduce difficulty of maintenance	Time required to follow methodology
Improve quality	Cost to follow methodology
Improve security	Not all techniques in the methodology may be applicable
Improve back-up and recovery	
Align IT department and EUC systems	

A study by Koh et al (1996) identified a lack of direct user participation and a lack of direct accountability in the design process. They attempted to solve the problem by developing the SOS-1 method (Systems for Supporting the Selection Process).

However, after an evaluation of the method, Koh et al (1996) found that there were some major defects with it such as:

(1) Not enough direct user participation in the design process. Future users are only asked to specify requirements or make decisions on various aspects of the system. Making decisions and formulating specifications for the system is important but it is not enough. The user should be a co-designer.

(2) No feasible accountability, since users are not familiar with the predominantly technical jargon and hence, cannot refer to the design process and ask questions or point out inconsistencies and misinterpretations. This may lead to inadequate transparency in the whole design process.

Greenspan (1994) states that a critical early activity is requirements definition, when the requirements for an information system are determined and expressed as a requirements specification. Curtis et al (1988) stated that if requirements specifications are incomplete, unclear or incorrect it may cause significant difficulties during subsequent development activities. This is why this area has proven to be so important and the need for effective communication is paramount. Newman and Roby (1992) state that the requirements definition is a collaborative effort that relies on effective communication and interaction between various stakeholders groups, managers, end-users and systems development professionals.

Darke and Shanks (1996) state that this communication between the developer and the user can help ensure that the system delivered meets the needs and expectations of all stakeholders involved. This can be done by facilitating understanding of their various viewpoints and requirements and supporting the management of conflicts and inconsistencies between them. However, Monk (1991) states that the specification of users' requirements is traditionally seen as the first step in the development of new IS and products. It is the initial part of the process of turning the demands (requirements) of users or clients into technical specifications for system design and development. Both the quality and usefulness of Information Technology (IT) products are critically determined by engineers success in eliciting requirements from their clients and in analysing and clarifying that information as the basis for development work.

Thomas (1996) states that two activities – requirements analysis and evaluation – traditionally seen as discrete activities in design, are now understood to be interleaved processes in the development of systems to support co-operative work: systems which themselves rest on (i) an understanding of the requirements the "social and organisational factors" imposed and (ii) the development of more encompassing measures of "elicitation". For requirements analysis – seen as a prerequisite activity in software development – the notion of "requirements gathering as a procedural and mechanistic process has moved towards a view of "requirements engineering" as the process of gathering "non-functional requirements" in design. Examining this area in more detail offers an opportunity to develop a process for carrying out requirements gathering. Kyng (1991) introduces co-operation as an important aspect of work for supporting computer application design to enhance quality of work and products. Therefore, it could be suggested that Kyng (1991) views requirements gathering as co-operation.

Vickers (1984) uses the term "negotiation" to describe interaction that may occur at any of the levels, but also argues that all communication involves a process of mutual persuasion which will probably change both parties, facilitating further higher-level interaction. He also defines five levels of communication in which interaction improves from a condition of open conflict (communication by "threat"), via levels of bargaining, request and persuasion towards a condition in which co-operation is dominant and the partners "engage" in joint effort to reach a common appreciation of the system, even where this includes an understanding of differences in the parties appreciation. He calls this the levels of "dialogue". Each ascending level, Vickers (1984) argues, requires each party to both understand each other better and to trust each other more. The development of such trust is vital to effective communication, but is easily destroyed. Within these scenarios, there are still likely to be different types of users and developers taking part in this communication process. Figure 2.15 presents a diagram indicating the type of process that might occur from interaction between user and analyst according to their respective roles in the situation, i.e. whether they are passive or active. This could mean that there are different co-operative elements in requirement gathering in different situations. It may also be true that once these elements have been established the process can be carried out more effectively.

Figure 2.15: Interaction of users and analysts in the requirements process (Brooks and Jones, 1996)

User(s)	**Active**	Impose requirements => Directive	Interactive requirements => Co-operative
	Passive	3rd party intermediary e.g. consultant = > Dissociative	Traditional requirements = > Captive
		Passive	Active
		Analyst(s)	

Finally, Yourdon and Constantine (1979) defined structured design as "the art of designing the components of a system and the inter-relationship between these components in the best possible way. They also define it as the process of deciding which components interconnected in which way will solve some well specified problem.

Orr (1981) investigated structured requirements definition and found that it is not always possible to elicit the required outputs from the user. He also states that it is necessary to develop a planning and definition phase that moves from an initial problem statement (symptoms) to correct definition of the problems and scope (system identification). Therefore, the stage that forms the heart of the study is the requirements stage.

2.5 Summary

Systems development methodologies may have many good points if used correctly. However, it is possible to identify gaps in the methods in relation to capturing the information that the methods diagrammatically represent.

Once the study began to examine the systems development methodologies in more detail, it was possible to identify gaps in relation to the consideration of the user. Tables 2.4, 2.5 and 2.6 show where the study drew some initial conclusions about the methods and their relation to capturing user requirements.

Even though some of the methods identified that the user must be considered, it became necessary to try to better define the process in order to add some structure so that the user would be able to participate effectively and also feel more involved in the process. There is also an argument that some of the methods had a great deal to offer the study in the form of different techniques, such as bringing users in the same room as the developer. Another factor was the diagrammatic representation of information received, which was learned from the methodology investigation.

The various methods and techniques have been identified in the literature. These methods have been proposed and investigated by a number of authors. Many identify requirements acquisition as being a very important stage if not the most important stage of the systems development life cycle (Reubenstein et al, 1991). However, Davis (1982) and Brooks (1987) have identified this stage as being difficult to complete correctly.

The area of requirements engineering has been investigated in this chapter and has summarised the work of Yu (1998), Potts (1997), Rolland et al (1998) and Plihon (1998). All have examined how goals are used within this area, which help the study to generate thinking patterns for carrying out this stage of the life cycle.

The work of Allen and Frost (1998) on use cases has formed some major assistance in the direction of the study, showing that it is possible to give the user a greater level of interaction, where the requirements of the user are the main focus. However, these cannot be left in isolation as interaction with the user does have to consider various problems such as abbreviations, ambiguity etc which Allen and Frost (1998) have also described.

The methods such as ADISSA and PAISLey offer a specific language structure to deal with requirements engineering. These help to deal with the data that is collected. However, the study wants to create an even smaller gap between the developer and the user.

Yadav (1981) also agrees that not enough time is spent on determining requirements. His study of the methods used in 1981 offered some interesting attributes to this research. The methods identified the flow of information linking organisational factors and formalising the process. However, once again it can be observed that there is not enough detail given on how the developer should interact with the user and putting the user in a position to specify their requirements.

Kyng (1995) has been able to offer a solution that is closer to the goals of the research, such as participatory design, but there is still confusion as to its effectiveness, due to no clear definitions of the level of user involvement in participatory design. Davis (1982) identified strategies for carrying out the process. These offer some options such as asking, deriving from an existing information system etc. This can still be argued as being led by the developer, but again there is no strategy for the process.

From reviewing the literature it became necessary to identify some questions that needed to be answered to establish where this research would contribute to existing work.

Page-Jones (1991) examined the structure of the process of involving users. Their work called out for a need to formalise the methods and identified the lack of use of structure as being a weakness on the part of the developer. Therefore, the study will then be able to ask 'would a more structured approach assist in carrying out the process of capturing user requirements?' The literature suggests a tendency to lean towards there being a need. However, Fitzgerald's (1997) work on the relationship between developer experience and methodology use, shows that this structured approach can vary according to the type of developer.

The area of user involvement has been well cited. Various writers have investigated this idea and the interest has shown the need to ask the question, should users be involved more? Once again the literature alone suggests that there is a need to involve the user more. As far back as 1975, Norton and McFarlan's work suggests that by involving the user more it would be more likely that the system would obtain a more accurate and complete assessment of user requirements. The work of Cheney and Dickson (1982) suggests that the user is becoming more critical to the success or failure of the system. Mumford (1997) asks why this type of design is not more popular and Taylor (1998) would like to know why the methods used are so complicated. This has helped to generate a need for more involvement. However, there is still a gap as to what is the result of more involvement. Yaverbaum (1988) examined this area and suggests that once a system has been implemented it would have been done as a result of a study to identify the need. This could therefore suggest that the use of the implemented system would result in the goals that have been identified before the system was implemented. One likely goal, is an increase in productivity. Therefore, the more the system is used, or once the system is used effectively, then there should be an increase in productivity. Yaverbaum (1982) suggests various factors that affect user satisfaction, one of which is participation; the work then suggests that once the user is satisfied then they should be motivated and therefore be more productive. Darke and Shanks (1996) identified that the relationship between the developer and the user would help meet the systems stakeholders expectations.

Once the research questions were identified, it became necessary to identify how the research would be carried out. This will be identified in chapter 3, which is the research methodology. One last pointer in this discussion relates to how the method will be developed. Gundagon (1995) states that there is not a clear-cut distinction between the structured analysis and design methodologies. He also states that it is believed that any terminology used is acceptable as long as it is reasonably well defined, prescriptive and consistent. Therefore, this study would adopt various ideas from what has been investigated in order to achieve a solution, which will then be exposed to industry and evaluated to identify how it could be used and where improvement can be made.

CHAPTER 3

RESEARCH METHODOLOGY

3.0 Introduction

The literature review (see chapter 2) was used to identify the weaknesses of existing systems development methods and especially methods designed to address user requirements. The conclusions suggested that there was a gap in the knowledge, where the interaction between the developers and the users needed to be addressed.

The existing structured methods for developing systems offered developers a guide that they could use to carry out the systems development process. Folkes et al (1992) described how the SDLC offered a structured approach from feasibility to implementation and maintenance. This trend of structuring the process continued with the development of methods such as SSADM and YSM, where not only a structure was offered, but also a tool that graphically represented the system. Further investigation into structured methods lead to RAD, where the user has been brought forward into a more prominent position. However, even further investigation lead to methods that claim to offer the user a more important role, such as Use-Case modelling, goal modelling and scenario authoring. According to the review of these methods there still seemed to be a problem with what previous methods had to offer in terms of their ability to involve users. Therefore, more information is required to build on the current information.

The research methodology needs to consider the source and collection of relevant data, the specification and development of a new method and its evaluation.

Methods such as data collection, analysis and evaluation are considered and discussed in this chapter to find a suitable research methodology to carry out the study. The work of Treacy (1985), who suggests that we should look beyond IS/IT research for guidance is considered. Yin's (1989) work is also considered. He suggests a similar option, encouraging the researcher to think further than the methods used before. Therefore, this research methodology discusses a collection of methods used in different fields of research. Table 3.1 describes six sections that the research should consider according to Morris et al (1987). This should help to form the rationale for the chapter.

Table 3.1: An Annotated Cognitive Domain Taxonomy (Morris et al, 1987)

Section	**Description**	**Relation to Research Methodology techniques**
1. Knowledge	Recalling information pretty much as it was learned. Knowing the major areas and methods of enquiry	Data collection
2. Comprehension	Reporting information in a way other than how it was learned in order to show that it has been understood	Research analysis Qualitative and Quantitative analysis Clustering Triangulation
3. Application	Use of learned information to solve a problem	Research analysis Specification of the new method of involving users
4. Analysis	Taking learned information apart	Research analysis Specification of the new method of involving users
5. Synthesis	Creating something new and good based on some criteria	Research analysis Development of the new method of involving users
6. Evaluation	Judging the value of something for a particular purpose	Research evaluation

3.1 Data collection

3.1.1 Selecting samples

Black (1993) states that how a researcher chooses a sample from a population will determine whether the members of the sample group can be considered to be truly representative of that population. He states that often population parameters with which to compare are not available and the researcher must depend upon the rigour of the sampling process to justify the representativeness of the sample. Kerlinger (1986) provides a definition of random sampling that states that it is a method of drawing a portion (sample) of a population so that all possible samples of fixed size *n* have the same probability of being selected.

Black (1993) has proposed a selection of criteria for evaluating representativeness. He has created a list which is as follows: (1) Whole population, where all findings apply to the whole population. (2) Random selection from a specified population, while there is no guarantee that the sample is perfectly representative, it is the soundest approach giving the highest probability that a sample is representative. (3) Purposive sampling from a specified population, where some attempt has been made to select a representative sample through specific criteria or characteristics related to variables that are to be controlled. (4) Volunteers, this will include a sample generated by accident, convenience etc. While there is some endeavour to obtain a sample that could be considered representative, such a sample is not very convincing. (5) Unidentified group, where the description of the sample or sampling technique is not sufficiently clear either to indicate the population or to justify any generalizability to a population.

Alternatively Moore (1991) has identified some categories where sampling can produce misleading data. Black (1993) states that these errors may not be a problem that is caused by the type of sample. Moore (1991) has recognised that there can be problems in obtaining the correct sample. This problem can be overcome by obtaining recognition for the execution of the study by higher management, which will hopefully ensure that the selected sample will be able to participate. The ultimate goal would be to use the whole population as suggested by Black (1993). A random selection from a specified population could be used this is due to the ability to access participants within companies. This method which is also suggested by Black (1993) is more likely to be achieved. The amount of users who will get involved will depend on the schedule of the company.

3.1.2 Observation, Interviews and Questionnaires

Robson (1994) identified some methods that could be used for obtaining results from the usage of the technique that was being tested. One of the methods is observation. He states that as the actions and behaviour of people are a central aspect in virtually any enquiry, a natural and obvious technique is to watch what they do, to record this in some way and then to describe, analyse and interpret what we have observed. The advantages of this method is its directness. He states that you do not ask people about their views, feelings or attitudes; you watch what they do and listen to what they say. He also states that this directness contrasts with, and can often usefully compliment information obtained by virtually any other technique. Interview and questionnaire responses are notorious for discrepancies between what people say that they have done, or will do, and what they did or will do (Oskamp, 1977; Hanson, 1980). However, Robson (1994) states that there is also the very practical problem with observation that it tends to be time-consuming. This technique has an advantage in that it allows the natural reactions of the participant to be recorded. If the participant is not aware that the observation is taking place then this may allow them to act more naturally.

Leedy (1985) states that all truth is not apparent by means of studying past records. We learn some truth by studying, through observation, the events that are taking place in the world around us. He also states that this type of research is also called the Normative survey. The name implies the assumption that whatever is observe at any time is normal and under the same conditions, could conceivably be observed again in the future. The basic assumption underlying such an approach is that given phenomena usually follow a common pattern.

According to Cohen and Manion (1989), an interview is a conversation 'initiated by the interviewer for the specific purpose of obtaining research objectives of systematic description, prediction or explanation. Robson (1994) states that an interview is a kind of conversation with a purpose. He states that interviews carried out for research or enquiry purposes are a very commonly used approach, possibly in part because the interview appears to be a quite straightforward and non-problematic way of finding things out. However, Powney and Watts (1987) have stated that such apparent simplicity is deceptive. They argue that it is 'as easy as writing a book – most of us have basic literacy skills but few attain literary art'. A point which may be considered in the development and carrying out of interviews is the bias associated with gaining the results. Gavron (1966) carried out research into the position and opportunities of young mothers, and was very conscious of the dangers inherent in research by solitary interviewers. She wrote ' It is difficult to see how this can be avoided completely, but awareness of the problem plus constant self-control can help'.

Watts and Ebbutt (1987) state that group interviews are attractive when the research involves studying an established group. This group could be represented by a user group or a group of developers. They also state, however that there is a clear disadvantage of using this approach. This is that it is difficult or impossible to follow up the views of individuals; and group dynamics or power hierarchies affect who speaks and what they say. An overt appeal for other contributions can help. This group situation may be of great importance to this study as the user-group may be used for gathering information.

These statements have shown that the interviewing process could be a useful method of gathering the information that the study may require to achieve its goals. However, Powney and Watts (1987) show that it is not a simple process, as the questions asked must be developed in a way that the answers are relevant to the research.

The process will involve interviews and questionnaires together with observation. This combination has been used by Oskamp (1977) and Hanson (1980) who used the interviews and questions for collecting the data but also used the observation for back up purposes as they state that people say that they have done, or will do things but there are usually discrepancies in this information. Therefore, the use of observation may prove to be useful. This will also be useful to identify the reactions to the use of the method, showing where the user puts more effort into the process. Giving an indication of how difficult or easy they find it.

The interviewing process will follow the work of Cohen and Manion (1989) who stated that an interview is a conversation initiated by the interviewer for the specific purpose of obtaining research objectives of systematic description, prediction or explanation and Robson's (1994) view that an interview is a conversation with a purpose. This may be the most appropriate thinking pattern for this process as workers usually feel more intimidated by a very formal approach.

The number of interviewees and the time factors are very difficult to specify as these will be dependent on the organisation. However, Whyte (1994) used a set of 10 individual 90 minute interviews to obtain data to form a study on understanding user perceptions of successful information systems. However, based on the survey of users and developers carried out within this study the process of interviewing users and getting them to fill out questionnaires lasted between half an hour to an hour. Therefore, the study would aim for an average of a 60-minute time slot (this will depend on the time that the participant is available) for each person who would be able to get involved.

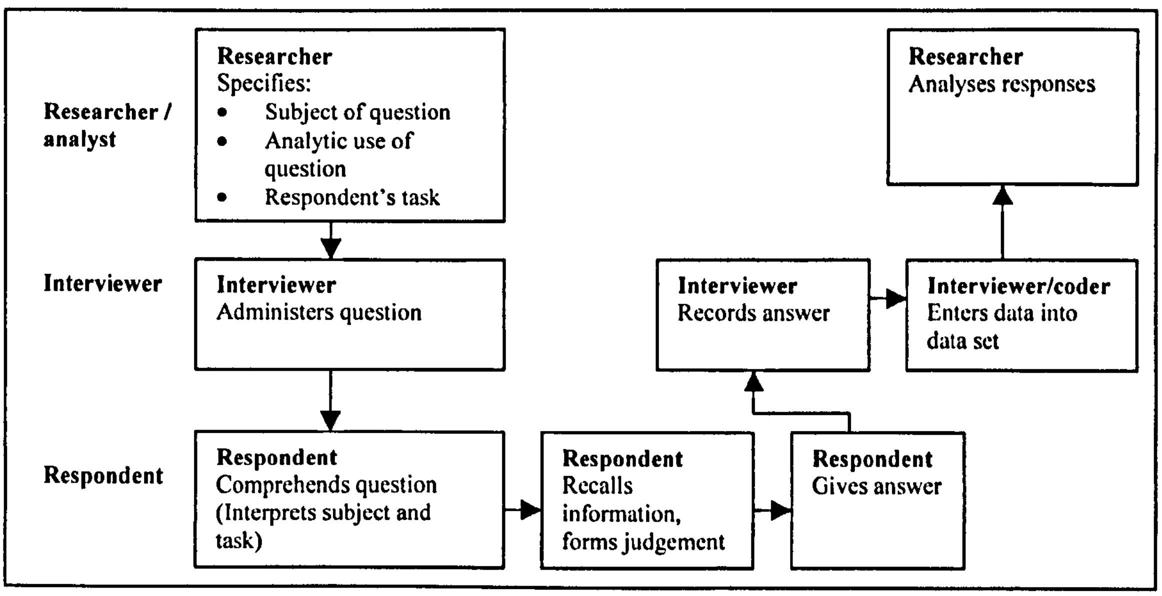
3.1.3 Questionnaire design

Berdie et al (1986) investigated the limitation of questionnaires and argues that some types of question formats are uniquely suited to mail surveys, whereas other types only work well with interviews. Rating scales that require people to rate items are more versatile in mail surveys than they are in interviews. However, interviews work more effectively than mail surveys when recall and recognition questions are linked together. This may be important in this type of survey as respondents are expected to voice their personal opinions on things that will occur within the interview. In this case the questionnaire is used to obtain feedback on how the user or participant feels about the method that they are using and will use in the interview.

Czaja and Blair (1995) state that the survey questionnaire is the conduit through which information flows from the world of everyday behaviour and opinion into the world of research and analysis; it is our link to the phenomena we wish to study. Oppenheim (1992) states that the questionnaire has a job to do: its function is measurement. But what is it to measure? The answer according to Oppenheim (1992) should be contained in the questionnaire specification.

Figure 3.1 demonstrates a process in which the questionnaire can be used and specifies the different players within the process together with their roles. Czaja and Blair (1995) also suggest that the researcher develops the survey question and specifies its analytical use. Then the respondent has to do four things: interpret the question, recall relevant information, decide on an answer, and report the answer to the interviewer.

Figure 3.1: Model of the survey data collection process (Czaja and Blair, 1995)



Oppenheim (1992), Czaja and Blair (1996) and Berdie et al (1986) offer some interesting guidelines for the development of the questionnaire. Oppenheim's (1992) work identified a pre-questionnaire plan and the others identified the factors to consider once the interviewer thinks that they have developed a reliable set of questions. These factors seem to form some logical steps towards the development of the questions for this study, due to their relation to the interview process rather than the mailed approach.

The questions used need to deal with the desired outcomes of the study. These are to identify general feelings about the methods that have been used before, the way they would like to see things happen and their assessment of the proposed method.

In the development of the questionnaire Tourangeau and Rasinski (1988) state that we should recognise that even opinion questions involve memory. If the respondent is asked for an opinion about something that the respondent has thought about prior to the interview, then the response is simply a matter of recalling that previous judgement.

Dyer and Bouchet (1995) used interviews to gain information about the opinion of users of the relevance of references received. Questionnaires were also employed, in order to gain basic demographic data and measure how users evaluated the information that was retrieved. This would be the most appropriate method to use as the users within the firm span from a high level such as management to a less experienced level such as shop floor workers. A suggestion would be to keep the interviewing process to a minimum amount of time especially as the factory (within BICC General Cables) is in operation around 24 hours per day. Therefore, the most appropriate amount of time could be based on the users job, for example the more they do the longer the process will take.

3.2 Research analysis

3.2.1 Qualitative and Quantitative Analysis

Winter (1991) gives an example of representing qualitative and quantitative research. A comparison is made of two reports of a game of football. The reports are as follows:

Quantitative	Qualitative
Wimbledon 0 Liverpool 0	There was more excitement in the Selhurst car park than on the pitch.

The two examples represent one quantitative result and one qualitative assessment of the game. In this instance, which would we consider more important – the result of the game? The points or the passion? This will depend on what we are interested in. If we are supporters we may be more interested in the result. If we were neutral spectators we would care more about the quality of the game.

Blalock (1960) states that we can count coins and notes to check how much money we have. We can use a milometer to check on distance and a measuring jug for volume. In each case, we have a measuring device, which can express variations in quantity in terms of an established scale of standard unit. The units by which we measure can be thought of as variables. The most important point is that for each of these variables we can confidently measure numerical differences in the values they can adopt.

Another distinction that is sometimes drawn between quantitative and qualitative methods is that the former produces data which is freely defined by the subject rather than structured in advance by the researcher (Patton, 1980). Miles and Huberman (1994) have identified various tactics that can be used to draw conclusions and test the validity of data. To make it easier to get a brief understanding of the tactics, tables 3.2 and 3.3 can be used. These tables show the tactics and their main uses or aims, which would make it possible to choose what is best suited to any particular research and to what the researcher is trying to achieve. Table 3.2 shows the tactics that are used for generating the meanings from the data and table 3.3 shows the tactics that can be used for testing or confirming the findings.

Table 3.2: Tactics for generating meaning. (Miles and Huberman, 1994)

TACTIC	USES/AIMS
1. Themes	Help the analyst see "what goes with what"
2. Seeing Plausibility	Same as 1
3. Clustering	Same as 1
4. Making metaphors	A way to achieve more integration among diverse pieces of data
5. Counting	Way to see what is there
6. Making contrasts/comparisons	Pervasive tactic that sharpens understanding
7. Partitioning variables	Helps to differentiate between data
8. Subsuming particulars into the general	Helps to see things and their relationships more abstractly
9. Factoring	Same as 8
10. Noting relationships between variables	Same as 8
11. Finding intervening variables	Same as 8
12. Building a logical chain of events	Finding how we can systematically assemble a coherent understanding of data
13. Making conceptual/theoretical coherence	Same as 12

Table 3.3: Tactics for testing or confirming findings (Miles and Huberman, 1994)

TACTICS	USES/AIMS
1. Checking for representativeness	Assessing data quality
2. Checking for researcher effects (on the case and vice versa)	Same as 1
3. Triangulating (across data sources and methods)	Same as 1

TACTICS	USES/AIMS
4. Weighing the evidence	Deciding which kinds of data are most trustable
5. Checking for meaning outliners	Can test a conclusion about a "pattern" by saying what it is not like.
6. Using extreme cases	Same as 5
7. Following up surprises	Same as 5
8. Looking for negative evidence	Same as 5
9. Making if-then tests	Ways of submitting theories to the assault of brute facts, or to race with someone else's theory
10. Ruling out spurious relations	Same as 9
11. Replicating a finding	Same as 9
12. Checking out rival explanations	Same as 9
13. Getting feedback from informants	See what the informants who supplied the original data think

3.2.2 Clustering

Tuffe (1990) states the we thrive in information-thick worlds because of our marvellous and everyday capabilities to select, edit, single out, structure, highlight, group, pair, merge, harmonise, synthesise, focus, organise, condense, reduce, boil down, choose, categorise, etc. and separate the sheep from the goats. It is possible to acknowledge that we have a very acute ability to put like with like together and separate things that are different. Miles and Huberman (1994) state that clustering is a tactic that can be applied at many levels to qualitative data: at the level of events or acts, of individual actors, of processes, of settings/locales, of sites or cases as wholes. They state that in all cases we are trying to understand a phenomenon better by grouping and then conceptualising objects that have similar patterns or characteristics.

An example of how clustering can be used to draw out the main factors for achieving objectives is from Davis (1959), in a study of the acts of cab drivers who were interested in receiving a larger tip. They clustered the information to find these methods:

- Fumbling in the making of change
- Giving the passenger a hard-luck story
- Making fictitious charges for services
- Providing a concerted show of fast, fancy driving
- Displaying extraordinary courtesy

This method is very similar to some work reported by Dey (1993) that looked at grouping data. He states that the development of a set of categories allows the data to be organised through a variety of different distinctions. Data within each category can then be compared. If necessary, further distinctions can then be drawn within each category to allow for a more detailed comparison of data within a set of sub-categories. Conversely, data assigned to different categories can be compared and interrelated to produce a more encompassing analysis of the data. This method has also been used by Whyte (1994) for a similar study which looked at understanding user perceptions of information systems. Here the method was used to cluster data ranked by users. Miles and Huberman (1994) also state that there is a long tradition of content analytic techniques dealing directly with issues of coding, unitising, and clustering qualitative data that can be very helpful, from Holsti (1968, 1969) and Berelson (1971) to Krippendorff (1980), Weber (1990) and Carley (1990).

Everit (1980) states that cluster analysis is a generic term for a set of techniques that produce classifications from initially unclassified data. However, the reasons for classification may differ from user to user – a possible reason for the existence of such a variety of clustering techniques. For example, Ball (1971) lists seven possible uses of clustering techniques, these are as follows:

(i) Finding a true typology
(ii) Model fitting,
(iii) Prediction based on groups
(iv) Hypothesis testing

(v) Data exploration
(vi) Hypothesis generating
(vii) Data reduction

The clustering techniques as stated by Everit (1980) are used in may fields of research where the researcher is faced with a great bulk of observations which are quite intractable unless classified into manageable groups, which in some sense can be treated as units. Morrison (1967) gives an example of the use of clustering in the field of market research. He suggests that a large number of cities are available that could be used as test markets, but due to economic factors testing must be restricted to a small number of these cities. If the cities could be clustered into a small number of groups such that cities within a group were very similar to each other, then one city from each group could be used as a test market. Everit (1980) also states that in some cases clustering techniques may be useful in shedding light on previously made hypotheses. This is one of the reasons for the use of this method. The aim of the survey of users and developers is to test what the study deemed to be a problem in the development of systems for users. The aim of interviewing, questionnaires and observation is to identify whether what has been initially investigated is true. Therefore, the study aims to find ways of improving what is currently happening, then develop a method based on what has been learned and finally test the method and try to establish how useful it is in terms of its strengths and weaknesses. Finally Aldenderfer and Blashfield (1984) states that there are five basic steps that classify all cluster analysis studies, these are:

(i) Selection of sample to be clustered
(ii) Definition of a set of variables on which to measure the entities in the sample
(iii) Computation of the similarities among the entities
(iv) Use of a cluster analysis method to create groups of similar entities
(v) Validation of the resulting cluster solutions

They state that each of these steps is essential to the use of cluster analysis in applied data analysis. These will be considered in the design process of the research.

3.2.3 Triangulation

Jankowicz (1995) states that you might find yourself using more than one method or technique, in combination, either because your design calls for it, or because you want to use the results from one method to cross-check the results from another. Kane (1984) represents archival review, questionnaires, interviews, and participant observation as potentially overlapping in scope, he states that "If you had to stake your life on which of these is likely to represent the most accurate, complete research information, you would choose the centre [of the overlap] in which you got the information through interviews and questionnaires, reinforced it by observation, and checked it through documentary analysis. Here, you are getting not only what people say they do and what you see them doing, but also what they are recorded as doing".

3.3 Research evaluation

Rutman (1977) states that evaluation research is first and foremost, a process of applying scientific procedures to accumulate reliable and valid evidence on the manner and extent to which specified activities produce particular effects or outcomes. He states that there are inevitably some constraints which necessitate compromises in the ideal design and data collection procedures, including among others: cost, administrative factors, legal requirements and ethical considerations. He also argues that too often evaluation research focuses solely on outcomes. From such studies, it is impossible to account for the results and similar efforts cannot be replicated elsewhere if the results are positive and avoided if the results are negative.

Fitz-Gibon and Morris (1987) state that in planning an evaluation, you will always plan to measure the experimental group, that is, the group (e.g. of students, classrooms, schools, employees, divisions) who receive the program which is to be evaluated. In this studies case the planning involves setting up interviews with relevant participants and ensuring that relevant information is obtained so that the proposed systems development method will have relevant advantages to address the needs of industry.

Ruddock (1981) states that research generally aims at adding to our knowledge and understanding of the world, or some part of it. It has always been recognised, however, that much of the knowledge accumulated by research has practical use, so that an increasing proportion of research effort is directed towards specific applications. He states that from cybernetics we can borrow the term 'feedback' and posit this as the distinguishing characteristic of evaluation. He also states that the processes of evaluation are to be based on the assumption, or demonstration, that what is observed and recorded manifests a degree of attainment of value-based objectives.

The methods mentioned in this area of the study will assist in the generation of the evaluation design. Each method provides some useful factors that may help to form conclusions about the proposed method.

3.3.1 NIMSAD (Normative Information Model-Based Systems Analysis and Design)

This area of the study demonstrates how methodologies can be evaluated. The first method of evaluation comes from a method called NIMSAD (Normative Information Model-Based Systems Analysis and Design). This framework has been developed from problem solving in industry, consulting practise and 'action research'. Jayaratna (1994) states that it can be used for evaluating any methodology, not just information systems methodologies.

The aims of the framework are to:

- Serve as a way of understanding the area of problem solving, in general.
- Help evaluate methodologies, their structure, steps, for, nature, etc.
- Help to draw conclusions

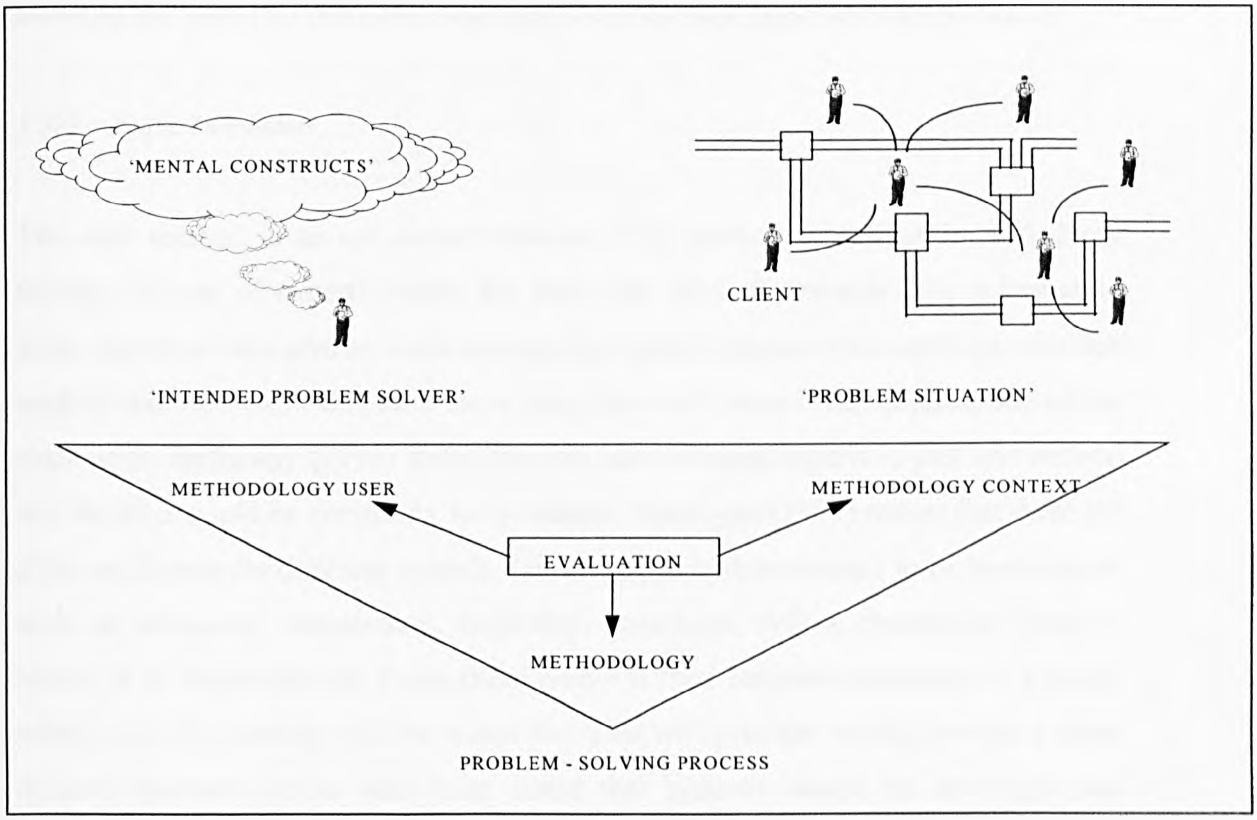
The NIMSAD framework separates the methodology that is to be evaluated into various sections.

There are four essential elements, namely:

- The 'problem situation' (the methodology context)
- The intended problem solver (the methodology user)
- The problem - solving process (the methodology)
- The evaluation of the above three

Figure 3.2 summarises these four elements of NIMSAD. The four elements would be useful in that they assist the evaluation process in ensuring that these specific elements are being addressed.

Figure 3.2: The essential elements of the NIMSAD framework (Jayaratna, 1994)



3.3.2 Information Economics

The principle of Information Economics is based on the view that there are measures of worth other than monetary measures (Parker et al, 1989). The method attempts to define a general 'utility' measure where 'utility' is defined as the satisfaction rating of an individuals revealed preferences. This is shown in the study by using the scales in the questionnaire. The method appraises the relative usefulness of different desired outcomes and attempts to rank them by applying a preference weight to each goal. McGolpin (1996) states that an advantages of this method is that it provides a mechanism for arriving at a decision by evaluating preferences and choosing the system which provides the highest satisfaction in terms of weighted preferences. He also states that the potential disadvantages are that the method involves a great deal of management time and it can be a costly exercise.

However, due to the qualitative nature of this study Information Economics may be used to highlight the non-monetary measures that the new method intends to address. This is in line with Parker et al's (1989) suggestion that the principle of the method is based on the view that there are measures of worth other than monetary measures.

3.3.3 Expert Opinion

The next method is to use expert opinion. This method of testing is carried out through the use of experts within the particular field of research. The information from experts should give an unbiased opinion of the research from someone who has studied and worked in this area for a long time and can see the implications of the work done. Bytheway (1994) states that this idea of using experts is just one method and the other could be circularity for goodness. Gundogan (1995) argues that there are different factors for different models (according to circumstances) to be investigated such as adequacy, consistency, reliability, structures, self - descriptive features, clearness to developments. From these points it then becomes necessary to identify which areas the strategy will be tested for. This will give the testing process a more defined approach. It has also been stated that systems should be developed and checked collectively with the people who are potential users and who are experts on the subjects (Perry, 1991, Bytheway, 1994, Dale, 1994, Qurashi et al, 1995).

3.3.4 Reliability and Validity

Bell (1987) states that Reliability is the extent to which a test or procedure produces similar results under constant conditions on all occasions. A clock that runs ten minutes slow some days and fast on other days is unreliable. A question that may produce one type of answer on one occasion but a different answer on another is equally unreliable. A question that asks for opinions may produce different answers for a whole range of reasons. Mehrens and Lehman (1984) state that reliability can be defined as the degree of consistency between two measures of the same thing. The two measures can mean a variety of combinations, for example: two different tests or measuring instruments, two halves of the same test, the same test or instrument applied on two occasions, two scorers using the same observation schedule. Wragg (1980) asks, would two interviewers using the same schedule or procedure get similar results? Would an interviewer obtain a similar picture using the procedures on different occasions? These are reasonable questions to ask yourself when you check items on a questionnaire or interview schedule. Bell (1987) and Bailey (1987) also state that there are a number of devices for checking reliability in scales and tests. These may include methods such as test-retest (administering the same test some time after the first), the alternative forms method (where equivalent versions of the same items in the test are given and results correlated) or the split-half method (where the items in the test are split into two matched halves and scores then correlated).

Rutman (1977) states that in considering reliability, the concern is how much of the variation in the measured phenomenon is due to inconsistencies in measurement, rather than in the phenomenon being measured. He asks the question "Can the measure be depended upon to secure consistent and stable results?" He states that there are a few measures which can cause differences in the results obtained at different times, which make the results have no logical pattern. Some of those stated are (1) respondent's or subject's mood, fatigue or motivation which can affect his or her responses; (2) observer's measurement, which can be influenced by the same factors affecting the subjects response; (3) the conditions under which the measurement is made, which may produce responses which do not reflect the "true" score; (4) problems with the measurement instrument, such as poorly worded questions in an interview; and (5) processing problems such as coding or mechanical errors. Black (1993) also looked at validity and states that basically, to ensure validity, any instrument must measure what it was intended to measure. This means that the instrument, as the operational definition, must be logically consistent and cover comprehensively all aspects of the abstract concept to be studied.

Anastasi (1990) and Cronbach (1990) have identified three types of validity which show some interest to the study. First Criterion-related validity, where data collected can be checked against an alternative set of data. They state that the researcher must consider the situation where it is necessary to check the validity of an instrument constructed to predict mathematical success: a set of results could be compared to the subjects' success in subsequent national mathematics examinations. Alternatively, results on a post-test of a training course could be validated by comparing them to on-the-job performance of the tested skills. The second is predictive validity, which checks how well performance has been predicted. An example would be the use of standardised aptitude tests which are an indicator of future performance. They state that in research this should be used on a representative sample of subjects in the population to confirm the criterion related validity of an instrument to be used on another, possibly larger sample. An example here could be a pilot study and then used within the main body of the research. The third type is concurrent validity, where the validity of the test is based upon knowing the present conditions of a sample of examinees or subjects.

3.4 Discussion

Within this study, the data is to be generated from questions and answers from interviews and questionnaires with users and developers. These answers will be brought together to form groups to find similarities and draw conclusions, this is similar to the clustering method, which has been investigated earlier in this chapter. McGolpin (1996) used this method successfully to form a study on an examination of the inter-related factors and issues affecting the degree of success with Strategic

Information Systems through the application lifecycle, where the objectives where to explain the degree of success with strategic systems and secondly to develop a framework that increases the understanding of the patterns and relationships that exist across the application lifecycle and how they impact success. These objectives are very similar to those that are to be achieved in this study. Where factors will be found to develop a method of increasing user involvement. Davis (1959), Dey (1993) and Everit (1980) also used this method to group and analyse data.

Miles and Huberman (1994) argue that much has been written about "triangulation" as a near-talismanic method of confirming findings. Stripped to its basics, triangulation is supposed to support a finding by showing that independent measures of it agree with it or, at least, do not contradict it. They state that the aim is to pick triangulation sources that have different biases, different strengths, so they can complement each other. Carley (1990) also suggests moving from uncontrolled to more controlled data types. This would be adopting a more qualitative approach which has been studied by Ellis (1993) and Wildermuth (1993) who have stated that recent research on end-user studies has made use of qualitative methods.

The evaluation methods mentioned earlier in the chapter offer some guidance as to what questions need to be asked in the interviewing process. The use of management levels to assist in the evaluation process offers the study a greater degree of validity. However, this is only possible with their co-operation. Table 3.4 demonstrates how the evaluation methods will be used, based on the writers who have investigated evaluation methods.

Aldenderfer and Blashfield's (1984) five basic steps that classify cluster analysis give an understandable guideline for the use of the technique. Which begins with the sample to be selected and then ends with validating the resulting cluster solutions. This will assist in the development of the clusters and for a better understanding of how to use the method.

Hsieh-Yee (1990) suggests that in the light and limitations of various data collection approaches, it was decided that a combined approach would be most likely to yield meaningful results. This could imply that the research would benefit from collecting data and analysing the information received using different methods which may prove to be more effective in the evaluation and testing process. This method which will be used to come up with the final conclusion is similar to the triangulation method.

The availability of resources has been mentioned to some degree with the work of Moore (1991). However, in order to evaluate the method, using the techniques mentioned earlier by Ruddock (1981), Jayaratna (1994) etc. It would be necessary to elicit the use of more experienced members or higher level staff within the organisation to answer questions that these methods use for evaluation purposes.

The idea of using such an approach would benefit this type of research, especially as there is no one method which enables the study to produce the desired outcomes. This leads to the development of the proposed method of carrying out this area which is shown in figure 3.3. Therefore, the final testing will be a process of reviewing the information gained from the users and then using the evaluation frameworks to answer questions posed by experts. This will lead to a further theoretical analysis which will incorporate measures for success to identify the reliability and the validity of the method. This keeps in line with the proposal to have a multi-method approach. As Mehrens and Lehman's (1984) work show, reliability can be defined by using different tests to measure the same thing.

Figure 3.3: Initial overview of methods to consider during the research

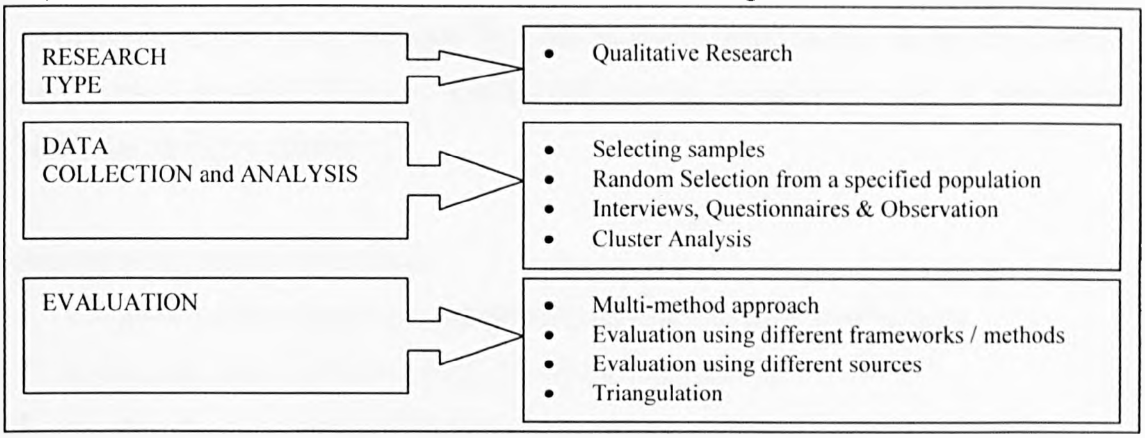


Table 3.4: Evaluation methods and their use within the study

Method / writer	**Useful considerations**
NIMSAD Jayaratna (1994)	Identifying problems
Information Economics Parker et al (1989) McGolpin (1995)	Understanding non-financial measures
Expert Opinion Bytheway (1994) Gundogun (1995)	Relevance within a particular field

Oppenheim (1992) states that the need for an appropriate research design arises whenever we wish to generalise from our findings, either in terms of the frequency or prevalence of particular attributes or variables, or about the relationships between them. This study aims to achieve a very similar goal to what has been stated by Oppenheim (1992).

3.5 The Research methodology adopted

The research method will be separated into four phases, each of which covers an area of the research. This will enable the study to have a better structure, to create a better understanding of how the new method was developed, and to show the validity of the investigation.

The aim of the research is to review the problems associated with IS that relate to user involvement and develop a method for capturing user requirements during the systems development process. This aim will be achieved by completing a set of objectives, which are stated in chapter 1.

Summary of research objectives:

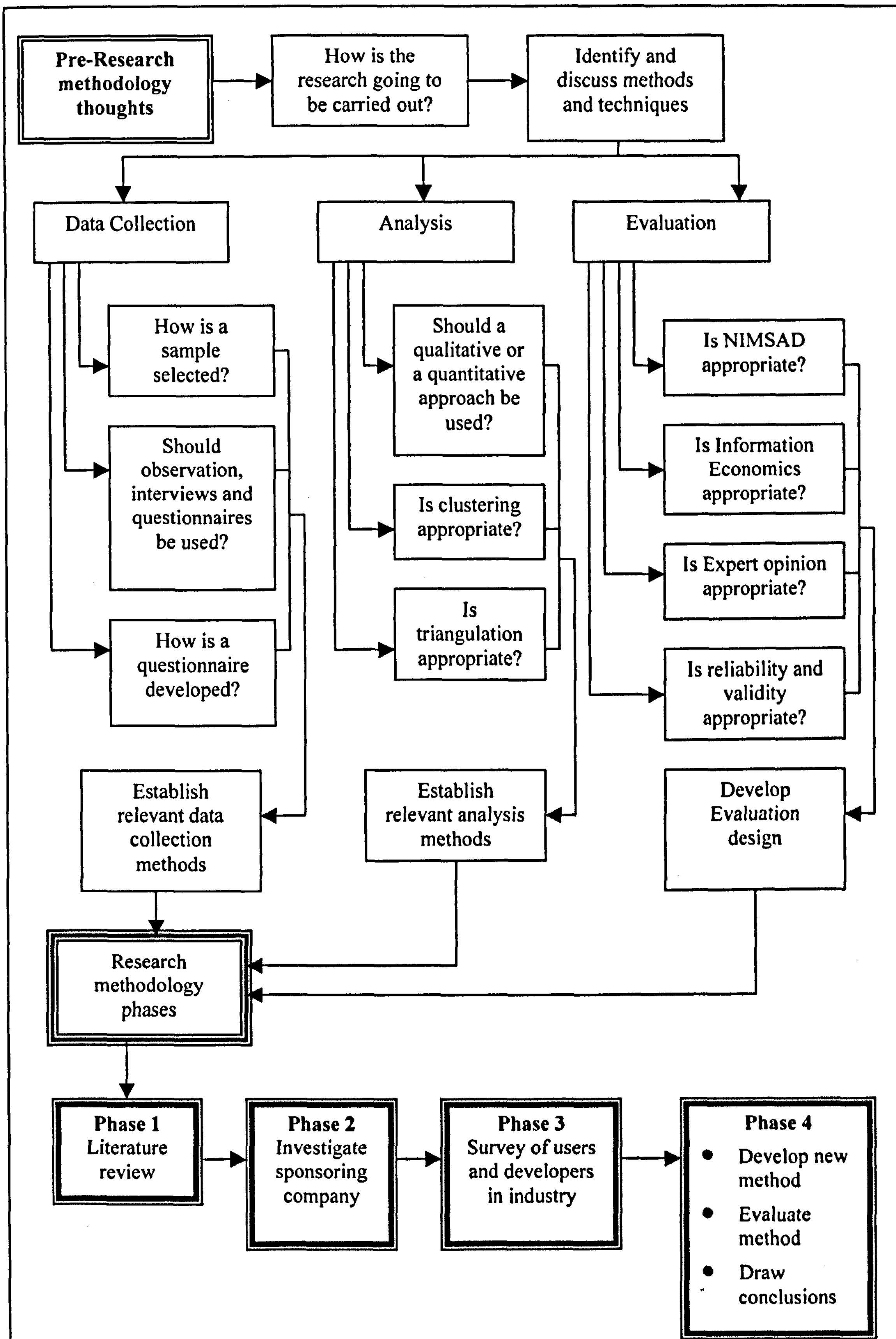
1. Compare existing literature and discuss relevance to user involvement.
2. Investigate user involvement and IS success and failure.
3. Identify the requirements for a new approach to capturing user requirements.
4. Develop new method considering previous objectives, evaluate method and show what the results indicate.

The research methodology (see figure 3.4) that is proposed in this chapter adopts the work of Gundogan (1995). Gundogan's (1995) approach is used to meet objective 1, as it defines a logical path to identifying the gap in the research area by reviewing the existing literature and then reviewing industry. This proposes a balance of industrial and academic requirements.

To meet objectives 2 and 3 various methods were reviewed and discussed in sections 3.1 and 3.2. The methods include interviews, questionnaires and observation, which could be used to produce both qualitative and quantitative data. It was important to clearly understand the needs of the people from the respect of the research. To obtain the information required, a survey approach appears best suited, using semi-structured interviews to capture the qualitative and quantitative results. These approaches are discussed in section 3.2.1.

Objective 4 was met by bringing the literature and the survey data together and forming a specification. This was used as the basis for the development of the new approach. In order to evaluate the new approach various evaluation methods were employed, such as, NIMSAD, Information Economics, and expert opinion. These methods are discussed in section 3.3.

Figure 3.4: The four phases of the research methodology



PHASE ONE

Literature Review

Identification and survey of the literature, which should highlight the need for the new method. This phase will also be used to establish the scope of the research identifying relevant areas to study in more detail. Here there will be a process of creating ideas to decide who will be the participants and what types of questions they will be asked. This is shown in figure 3.5.

PHASE TWO

Investigate sponsoring company

This stage will be used to gain an initial industrial perspective on the study, where results can be achieved to back up the literature survey. The ideas created in phase one will be tested with BICC General Cables. This will be used to see what types of answers are gained from the specified questions. This will lead to the development of a set of questions for the participants identified. This is shown in figure 3.6.

PHASE THREE

Survey of users and developers

At this stage the questions for the participants will be established together with the companies that are relevant to the study. This will then mean that the interviewing process can be carried out. Here results, plus the key information from the participants should be gathered at the end of the phase. This is shown in figure 3.7.

The main purpose of the survey is to identify whether there is a need for a new method of capturing user requirements from an industrial perspective. Once this need is established, the survey will be used to obtain information that will assist the development of a new method.

The objectives of the survey are as follows:

1. To develop an industrial perspective on user involvement in the development of information systems.

2. To identify an industrial perspective on the usage of structured methodologies.

The survey should addressed the objectives listed above. This will be achieved by interviewing users and developers of various companies. The companies that will be involved in the survey should consist of volunteers from a list of ex-Cranfield students (These students now work for various organisations. Therefore, they would represented the contact point within each company). Letters will be sent to the ex-students explaining the purpose of the research. Those who are interested will voluntarily take part in the study and interviews will be set up with users and developers within their organisations. These companies should all have implemented information systems. They would all have gone through a process of making changes to their current systems within the last 6-12 months and consist of developers and / or users.

The interviews will be structured, where each question asked will be listed in a questionnaire. The questionnaires that will be used is shown in appendix B (user questionnaire) and appendix C (developer questionnaire).

In order to analyse the information received from the interviewees the questions and answers will be grouped forming clusters. These clusters should made it easier to draw conclusions about the information that is received, by allowing small groups of answers to be dealt with separately.

PHASE FOUR

Development, Evaluation and Conclusions

The information gained in the earlier phases of the research methodology will be used to develop the new method of involving users in the systems development process. An evaluation is also carried out in this phase to justify the relevance of the solution to the problem stated in chapter 2 (literature review). At the end of this phase, conclusions will be drawn from the results of the study. This process is shown in figure 3.8.

From the previous discussion in the literature review (see chapter 2) it has been concluded that there is a need for a new approach to involving the user in the process of information systems development. This new method would have to be understood by developers and users and should have some element that brings them closer together helping them to understand each other's requirements.

The area of information systems development was studied first. This was very strong on the need for a structured approach. This approach has evolved due to haphazard systems development projects ending up going over specified dates and resulting in system failures. Therefore, in order to achieve the structured approach various systems development methodologies were investigated to identify their considerations towards involving the user and identifying their strengths and weaknesses. The methodologies that are available for use in industry such as SSADM, RAD, DSDM etc. were analysed.

The specification for the proposed method will be developed by bringing together the two sources: (i) literature and (ii) survey of users and developers. From the two sources, factors were identified by their relevance in relation to filling the gap that had been stated by the literature. It was important to understand the subject area in depth and draw conclusions from the literature to show where there was a need for improvements. The survey of users and developers will be used to find out what is happening in the companies who are developing information systems. It should also find out if they have any problems and if they recognised the need for a new approach. The questions asked relate to the users level of satisfaction with the systems that they are currently using, the methods that are used to develop the systems, the problems that they encountered and the requirements of a new method.

Once these factors are found they will then be discussed further to see how they would be relevant in their ability to solve the problems that had been identified in the literature and the survey of users and developers. Then factors will be considered to form groups where the most important elements are discussed to form a final specification for the development of the proposed method.

Galliers (1992) identified a number of research methods in common use in the study of IS/IT. These include, Laboratory experiments, field experiments, surveys, case studies, theorem proof, action research, interpretive techniques. However, it is not clear what type to use when trying to achieve varying objectives. Vitalari (1984) has developed a table that related the research objectives to the research method that should be employed. This is shown in table 3.5.

Perry (1991) argues that as well as interviews, questionnaires are also used between major modelling events as part of the justification or validation process. This research will not result in the development of a model, but it will use information to develop a structure which is very similar in that it is an approach to improve a specific process. However, Czaja and Blair (1995) state that a distinguishing property of surveys is that the resulting data is often put to both descriptive and analytic use. They also state that on the most basic level, the idea of a survey begins with the desire to know some unknown characteristic of a population.

Table 3.5: Research approaches and Research Objectives (Vitalari, 1984)

Research objective	Research method employed
1. To improve the effectiveness of information systems in practice	Laboratory or field experiment
2. To study information systems failures or implementation errors	Survey or case studies
3. To study the impact of Information Technology and Information Systems on organisations	Survey or case studies
4. To study the role and effects of Information Technology and Information Systems on Society	Survey or case studies

After reviewing the work of Galliers (1992) and Vitalari (1984) it is suggested that a survey approach together with a field experiment should be adopted. The survey approach encourages responses from a collection of participants and the field experiment gives users the opportunity to interact with the new approach. This will allow the new approach to be tested with real data. This combination of events would imply that this area of the research would use a triangulation approach, which is suggested by Jankowicz (1995). This will be used as the basis for drawing conclusions.

Figure 3.5: Research Methodology phase one

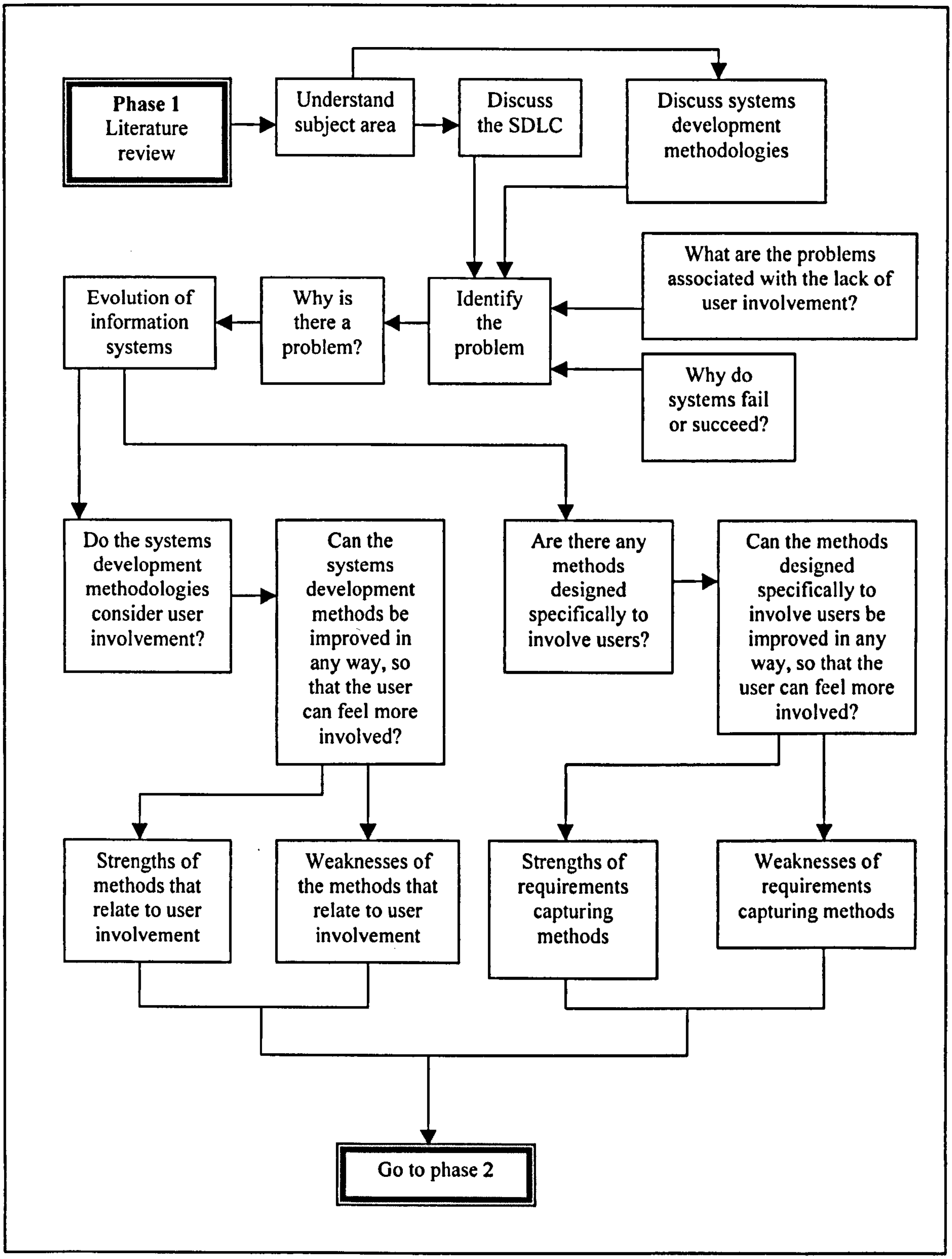


Figure 3.6: Research Methodology phase two

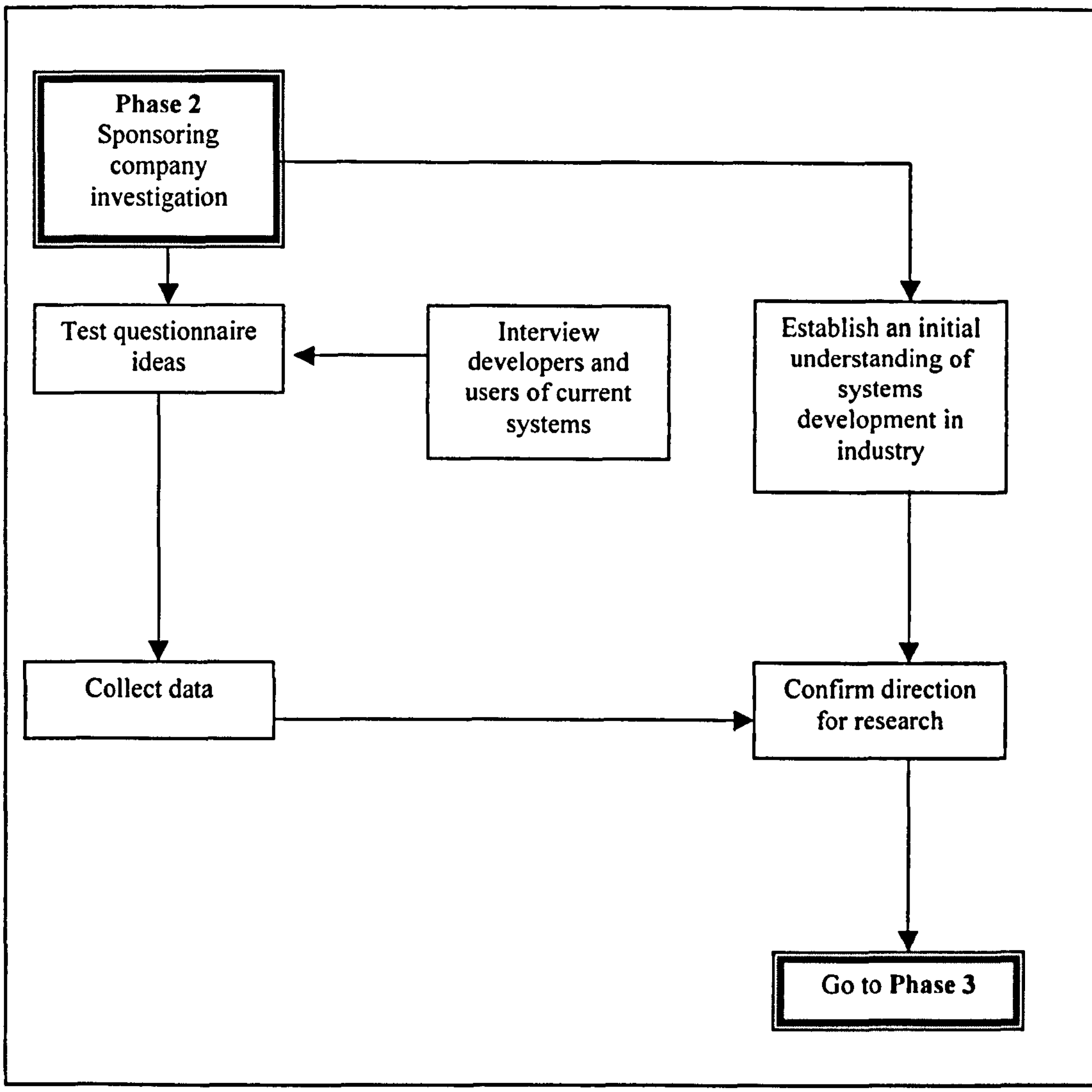


Figure 3.7: Research Methodology phase three

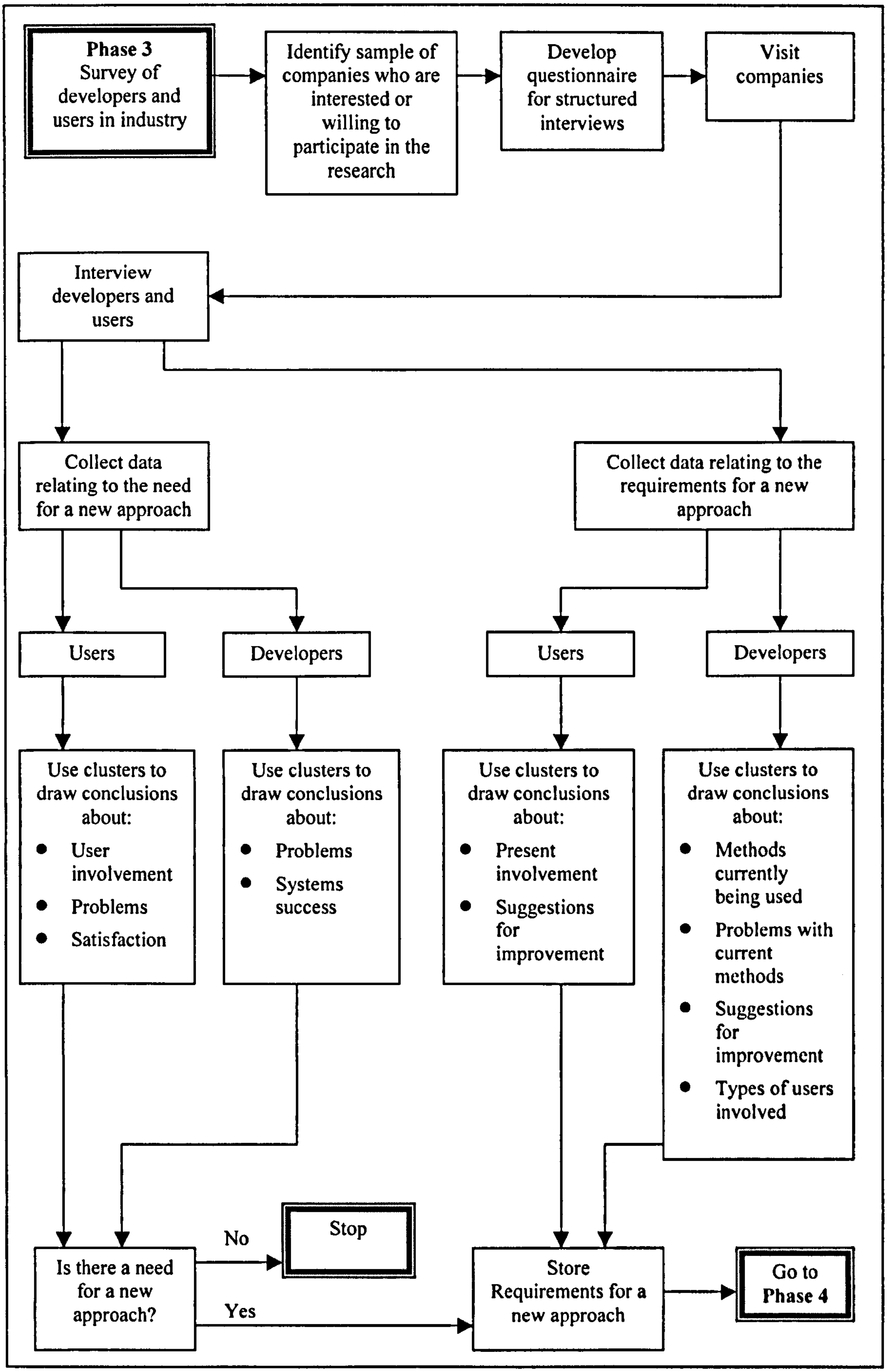
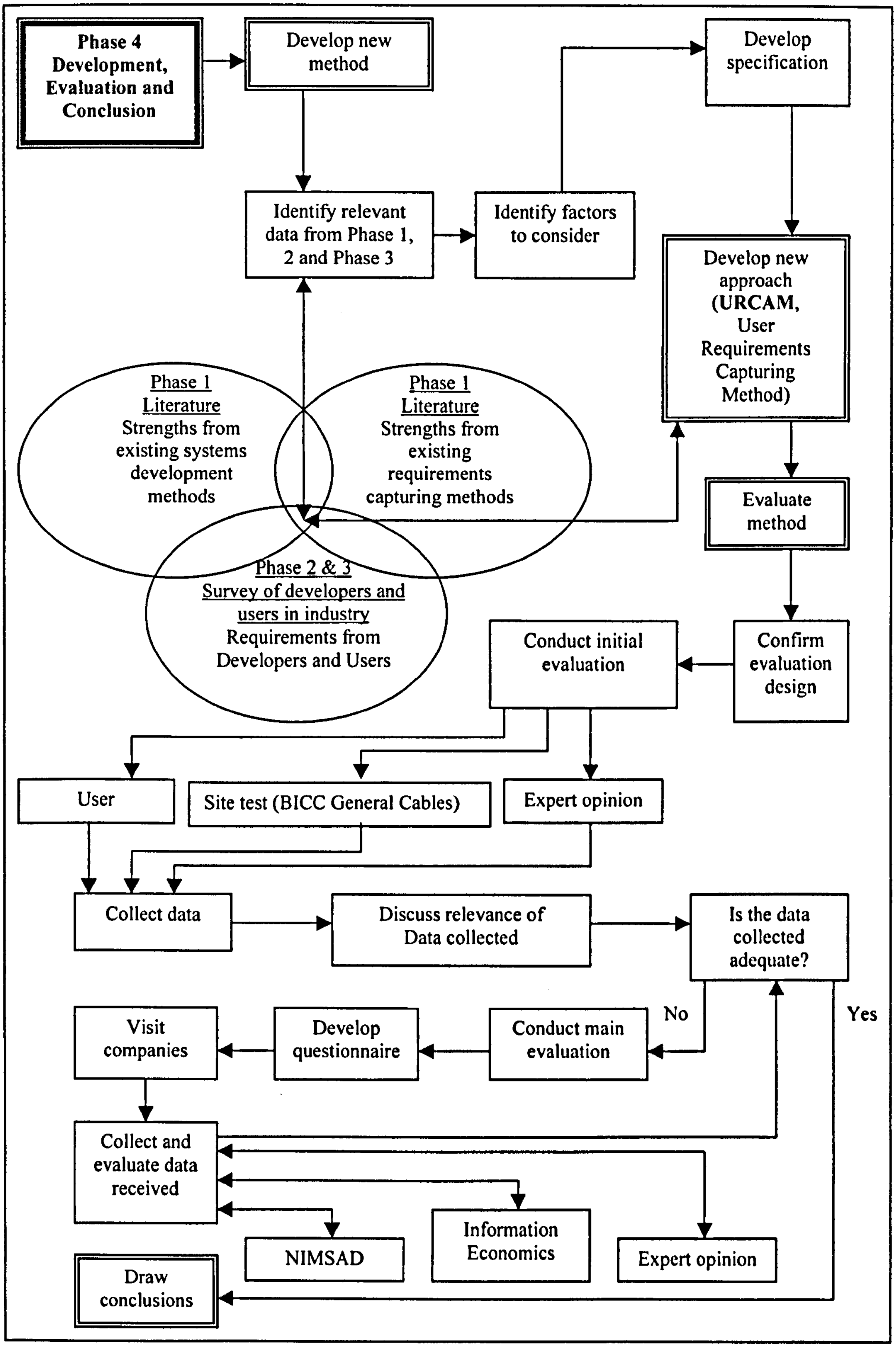


Figure 3.8: Research Methodology phase four



CHAPTER 4

SURVEY OF USERS AND DEVELOPERS

4.0 Introduction

This chapter presents the results and the analysis of the survey of users and developers. This survey is based on an investigation of user involvement in the sponsoring company (BICC General Cables) and a further investigation into a number of companies (see table 4.1), where the following objectives are achieved through the use of questionnaires (see appendix B and C) during face-to-face interviews:

The objectives of the survey are as follows:

1. To gain an industrial perspective on user involvement in the development of information systems.

2. To gain an industrial perspective on the usage of structured methodologies.

Table 4.1: List of companies involved in the survey of users and developers

COMPANY NAME	DISCIPLINE	USERS	DEVELOPERS
BICC GENERAL CABLES	CABLE MANUFACTURER	4	2
ROVER	CAR MANUFACTURER	2	1
SKANDIA GROUP	INSURANCE / FINANCIAL MANAGEMENT	0	3
SITEL CORPORATION	TELEMARKETING	1	1
ALFRED BERG UK LTD	STOCKBROKING	1	1
RAF BRAMPTON	ROYAL AIR FORCE LOGISTICS	1	1
MCCOORCODALE ENVELOPES	ENVELOPE MANUFACTURER	1	0
FORD (ENFIELD)	CAR COMPONENT MANUFACTURER	1	1
REXAM PACKING	METAL TRAY MANUFACTURER	2	1
FOSROC	CERAMIC MATERIAL MANUFACTURER	1	3
DAGNELL ELECTRONICS	ELECTRICAL COMPONENTS	0	1
AIR PRODUCTS	LARGE INDUSTRIAL CYLINDER MANUFACTURER	0	1
IBM	COMPUTER HARDWARE & SOFTWARE SPECIALISTS	1	0
BRITISH AEROSPACE	MILITARY AIRCRAFT MANUFACTURER	2	1
LOGICA	IT SPECIALISTS / DEVELOPMENT	0	1
	TOTAL	17	18
	TOTAL RESPONDENTS	35	

4.1 Investigation of user involvement in sponsoring company

4.1.1 Introduction

The purpose of this investigation was to use the sponsoring company (BICC General Cables) to confirm the direction of the research by interviewing developers and users to try to find out what their views were on user involvement. This phase was also used to gain some industrial experience to establish an understanding of systems development in industry. The interviews with the personnel involved were conducted with the use of a questionnaire. The questions used would be tested in this investigation to determine the types of responses that were generated from the questions. Once the questions initiated responses that were relevant to the study then they would be used to conduct a further survey of users and developers (see phase 3 of the research methodology). The developers and users who were involved have all been involved in a systems development project before. Therefore, they would be aware of what happens and would also have views regarding the methods used. This part of the research also included an introduction into the working environment which was very important to understand.

4.1.2 Findings

This investigation was carried out over a three week period within the MSE (Manufacturing Systems Engineering) department, where the researcher spent time working with the customer interface engineering team getting to grips with the processes being carried out in the BICC General Cables Wrexham factory.

Six structured interviews were conducted during the investigation. These interviews were with two developers and four users who had experience of systems development projects. Some of the problems identified are listed below:

- The specifications available on the system are not being used to their fullest potential.

- Difficult to maintain old systems as developers have changed, technology has changed and user requirements are more sophisticated.
- The users of the system do not seem to have been trained well enough. This may be the cause of the lack of use of the system.
- It is difficult to get access to the users that you want at the right time to be involved in the development of new systems.
- Improvements to the systems usually depends on a new need. However, new technology can lead to changes in the system that enables new methods to be used.
- The systems support staff and administrators should know how the system is designed and structured. However, the users should not really need to know what is involved in the development process.

The Customer Interface Engineers (CIEs) state that a successful system is one that meets the needs captured from the users. This information can be obtained by using questionnaires that are given to users after a system is produced. This will identify various opinions about the system and its use by the users. The system should be flexible enough to be changed where and when it is necessary.

At present a feasibility study is produced. This is backed up by further research into what the system is supposed to achieve, which can be carried out through meetings with other engineers. Users normally have problems with the system produced and sometimes within one month of the implementation the users may require changes.

In the past there were problems but they were not resolved. This may be a reason for the lack of interest in the process of identifying the needs of the system. The right person may not have been spoken to, this may be the cause of the problems in the specifications. There are no 100% guarantees in systems development. However, there is usually a feeling of confidence that the system requirements have been identified correctly.

The users of the systems were generally unhappy about the current systems. The main problem was the reliability of the current systems. However, there were some conflicting results as management levels suggested that they had more control, which they were happy about and operators suggested that they had less control, which they were unhappy about.

When the users were asked about the levels of involvement that existed, they stated that it was low and should be improved. They suggested that there was no pre-planning or structured approach. The process of capturing the requirements consisted of developers visiting users while they worked and asking them questions about the operations that they carried out.

The users felt that they would require more interaction to make them feel clearer about what is going on in the process. They also felt that once they were involved they would feel more enthusiastic about using the systems. Finally they stated that there seems to be an attitude that if the problem does not seem large in the eyes of higher management and the development team, then the problem is not important and is not dealt with promptly.

4.1.3 Conclusion

This investigation tested the questions for a further survey and established an understanding of systems development in industry. The findings suggested that there are problems with the interactive process between systems developers and users. Therefore, there is a need to develop a method that would encourage the user and developer to interact in a more structured format. However, this would be investigated further with a range of other companies to justify whether there is a need and also to identify the requirements of a new method once the need has been established.

4.2 Establishing the need for a new approach

4.2.1 Users perspective

Figure 4.1: Cluster titles to form evidence from a users perspective

CLUSTER 1: USER INVOLVEMENT

CLUSTER 2: PROBLEMS

CLUSTER 3: SATISFACTION

In order to establish a need for the research from an industrial perspective, users were asked questions to identify their views about user involvement. Figure 4.1 illustrates the titles of the clusters that were used to identify a need for a new approach from a users perspective. The clusters contain groups of questions that were in the questionnaire that was used to interview the users.

Cluster 1: User Involvement [Users]

The questions and responses that were used to draw conclusions relating to user involvement are shown in table 4.2.

Table 4.2: Table illustrating evidence relating to user involvement

USER INVOLVEMENT		
No.	**Question**	**Response data**
1	Would you like more interaction?	76.47% YES – 23.53% NO
2	Would you like less interaction?	0% YES – 100% NO
3	Would the fact that you were involved increase your willingness to use the system?	94.12% YES – 5.88% NO

USER INVOLVEMENT		
No.	**Question**	**Response data**
4	By what percentage do you think that user involvement should be increased?	Average increase of 51%
5	By what percentage do you think that user involvement should be decreased?	Average decrease of 0%

The first points that stand out are that 0% of the respondents said that they would like less interaction, and 0% said that they think that user involvement should be decreased (see table 4.2). This suggests that we should not decrease the levels of user involvement. According to the literature review in chapter 2, conclusions have been drawn to suggest that there is a need for more involvement between users and developers. In response to the question "would you like more interaction?" 76.47% responded "yes" (see table 4.2). Users were asked "by what percentage do you think that user involvement should be increased?" the average increase was 51%. These results suggest that there is a perceived need for an increase in user involvement. However, to add to this evidence an additional question was also used to help to understand users views on involvement. The question "would the fact that you were involved increase your willingness to use the system?" was also asked, the response given was 94.12% of the respondents responded "yes" and 5.88% responded "no". These responses suggest that the users involved in the survey are not content with the amount of involvement that they are currently receiving during the development of information systems. The conclusion is that they would prefer an increase in the level of involvement during systems development and this would result in an increase in usage.

Cluster 2: Problems [User]

Cluster 2 looks at the problems that are currently being faced within the companies in relation to their information systems. The questions and responses are show in table 4.3.

Table 4.3: Table highlighting the problems currently being experienced by users

PROBLEMS		
No.	**Question**	**Response data**
1	Are you happy with the systems currently in place?	29.41% YES– 70.59 % NO
2	If No, what is wrong with the systems?	• Tailored for a particular time – Not considering future needs • Not much structure for change or improvements • System is old and slow • Used old language to write the system specification • System slows down the flow of work • Creates errors • Functions that are expected are not found • Not user friendly • Problems have caused users to find ways around the system • Needs the ability to manipulate and analyse data easier • Reports can only be created at particular periods • Had to keep own records • Delays in getting the right information at the right time • Not integrated well enough • The data being generated is not trusted

PROBLEMS		
No.	**Question**	**Response data**
3	Is there adequate functionality in the system for your job within the company?	58.82% YES – 41.18% NO

From the responses given to the question, "are you happy with the systems currently in place?" 29.41% responded "yes" and 70.59% responded "no". Therefore, the first conclusion is that users are generally not happy with the systems that are currently in place. In order to identify whether this response has any direct relationship to the functionality of the system, users were asked "is there adequate functionality in the system for your job within the company?" the results in table 4.3 show that 58.82% of the respondents responded "yes" and 41.18% responded "no". This was not expected as in table 4.3 only 29.41% of the respondents said that they were happy with the current systems.

When the users were asked "what is wrong with the systems?" The list of complaints in table 4.3 show that the users cannot find the functionality. This relates to the question "is there adequate functionality?" The reason for the response given in question 3 on table 4.3 may be because the functionality is there but they cannot find it. After reviewing the list of responses to the question "what is wrong now?" users seem to have a lack of confidence in the system. The respondents say that "the system is not user friendly", and "they cannot manipulate the data the way they need to". These problems can be linked to the process of involving the user. The systems specifications should be agreed with the user and therefore these problems should not be an issue. The conclusions that are draw from this cluster is that the system has not been designed with adequate user involvement. Users are generally unhappy with the systems and have quite a few problems using it.

Cluster 3: Satisfaction [User]

Satisfaction in this case relates to the users personal feeling about their use and expectations of information systems in relation to their job. The results from this cluster are shown in table 4.4.

Table 4.4: Table showing the users views on satisfaction being experienced by end users

SATISFACTION		
No.	**Question**	**Response data**
1	What does satisfaction mean to you?	• Using something that helps you to deliver what you want • Using something that is easy to use • Easy to manipulate data • Being given good initial training • System that does what you want it to do • Knowing that there is support • Knowing how to deal with and identify problems • Jobs being able to be completed efficiently • Adequate speed of delivery of information • Reduction of wasted effort • Being able to trust that information is correct • Easy to visualise problems • Good explanation of why things are being done • Minimum input • System helps to do the job
2	What levels of satisfaction are you experiencing now?	Average satisfaction level of 50%

The results obtained from table 4.4 represent what users feel would make them feel satisfied and it also shows what level of satisfaction they are experiencing now. The users were not very clear about what they considered to be the issues that caused them to be satisfied. However, they were able to identify some ideas. The points that are closely related to the system are that achieving satisfaction means that the system must be able to deliver the required information and the information needs to be manipulated easily. Support and speed of delivery of information are also important factors. The main point is that the system should help them to do their job correctly.

The levels of satisfaction that are experienced state that the average level is 50%. This suggests that there is a balance of satisfied and unsatisfied users. This percentage raises another question, which is "how can such a high percentage be satisfied?" According to the results shown in tables 4.2 and 4.3 users are generally not happy with the systems that they are currently using, but the users are content with what they have. This may be because they are not aware of methods that could be used to improve the current situation. The results show that they would be happy if the levels of involvement were increased. However, they are not showing great levels of satisfaction. This not only relates to user involvement but also expectations regarding system functionality and user friendliness.

These suggestions can also be used to generate requirements for a new approach. Many of the suggestions, for example, using something that helps you to deliver what you want or using something that is easy to use are important factors in systems development. These suggestions consider the deliverables of the system and its usage. Therefore, the suggestions demonstrate that the user is both concerned about the results that the systems produces and also how easy it is to obtain the information that is required.

Support has also been mentioned. This means that the user would require a contact point to go to if there are problems with the implemented system. Rather than a company (e.g. software house) just developing a system, implemented it and then going away without offering support. There would be a requirement for that company to give the customer a contact point where they can obtain assistance with any problems and improvements to the system.

4.2.2 Developers perspective

Figure 4.2: Cluster titles to form conclusions from a developers perspective

CLUSTER 4: PROBLEMS

CLUSTER 5: SYSTEMS SUCCESS

Figure 4.2 shows the cluster titles that where used to draw conclusions from the developers responses regarding whether there should be any changes in the way that users are involved. The developers were asked two sets of questions. The answers are used to draw conclusions about the factors that developers consider before and after the implementation stage.

Cluster 4: Problems [Developers]

Table 4.5 lists some of the problems that developers have encountered. We can see that there is a variety of problems which developers are facing. These problems seem to be due to bad systems development practices and a lack of consideration placed on the needs of the business. After speaking to the developers it became clear that their aim is to produce what they can with the resources that they have available. This suggests that the resources that they are given are not sufficient for what they are trying to achieve. The developers should be encouraged to address this problem with their superiors rather than producing systems that are not satisfying the needs of the companies information systems.

The main points that relate to user involvement are that there is a bad relationship created between developers and users, incorrect data is being produced, there is a lack of knowledge about the systems and there is no ongoing support.

The problems listed in table 4.5 suggest that the systems currently in use are likely to fail. Once the information received from users is not correct then the system is likely to produce incorrect data.

Table 4.5: Table showing the major problems that developers are facing

PROBLEMS		
No.	**Question**	**Response data**
1	What are the main problem areas of the systems that are currently in place?	• Not delivering the business functionality • Too much integration • Too many checks necessary for a change to be made • Lack of resources to do what users want • Untidy database • System is old • Time constraints • System not taking into account users of the system • Lack of trust in the system • Bad relationships created between developers and users • Incorrect data being produced • Previous documentation needs updating. • Developers need to be closer to see what is going on with the system • System is too large • Lack of knowledge about the system • No time for method development • No ongoing user support • Outdated systems

One of the reasons why the relationship is bad between the developers and users is because in some cases developers feel that they know what the user wants. Evidence of this is shown by developers not having enough knowledge about the system (see table 4.5). The developer goes ahead and develops the system without adequate user input and then does not support the user after the implementation. This means that users have to calculate figures without the use of the system.

Cluster 5: Systems Success / Developers /

This cluster is used to form a group of questions that show what affects the success of the system before and after the system has been implemented. Table 4.6 shows the information that developers have suggested that relate to success.

Table 4.6: Table showing the responses relating to information systems success

SYSTEMS SUCCESS		
No.	**Question**	**Response data**
1	What methods are used to determine the final success that the system will produce before the implementation	• Periodical tests – functional test • No formal process • Test the whole system • Test for business processes • User acceptance testing • Full system test, including unit test, functional test, prototyping-which involves the user • Developer and user test for correctness • No methods • Users will go through the functional spec • Dummy tests are carried out on functionality • None, the system is implemented and then they see what happens • Sometimes there is a post implementation review

SYSTEMS SUCCESS		
No.	**Question**	**Response data**
2	How is the success of the system measured?	• Value for money • Delivery of functionality • User satisfaction • No real measure • Post project reviews • IT survey is carried out. This asks questions about how the system supports the user • Look at if the system meets the initial benefits required • See if the user gets what they require • Reduction in stock levels • Longevity of the system • System should allow company to detect and deal with problems easier • No negative impact on the organisation • Is the system easy to use • How much money does it save the company
3	How do you know that the information obtained from the users is correct?	• Normally trust the users to have said the right things • Users are walked through the requirements with the developer • Requirements are signed off by the users • Try to get a consensus opinion • Use dry runs where information is input into the system • Cross referencing with other users • Trial and error • Users must prove what they say is correct

The responses given in table 4.6 suggests that the developers are theoretically quite sound. They seem to know what is required from the system and how they should test the system to ensure that it does what it is supposed to do. However, if we look back at table 4.5 we can see that even with these measures and tests etc. the companies are still having many problems with their systems.

After reviewing the results shown in table 4.6 it is clear that the companies have some valid suggestions for checking that the information that is collected is correct. The results also show that the developers have a reasonable level of trust in what the users have said. This is a good point in relation to the user / developer relationship. However, even though these factors are true, users are still unhappy with the systems that are produced by the development teams.

4.3 Development issues

4.3.1 Users perspective

Figure 4.3: Cluster titles for information relating to development from a users perspective

CLUSTER 6: PRESENT INVOLVEMENT

CLUSTER 7: SUGGESTIONS FOR IMPROVEMENT

Figure 4.3 shows the two cluster titles that form the responses that were identified to assist the development of the proposed method. Each question highlights a factor that the proposed method can adopt or improve on.

Cluster 6: Present involvement [Users]

Table 4.7 shows the information that is used to draw conclusions about the methods used for capturing user requirements. All of the users interviewed have taken part in a development process. The fact that users have been involved before made it easier to identify how the process was carried out. This also helped to identify what users feel are good and bad points according to their experiences, so that improvements can be made.

Table 4.7: Table showing the users responses to how developers and users interact during the systems development exercise

PRESENT INVOLVEMENT		
No.	**Question**	**Response data**
1	How did the process take place?	• Meetings with developers and users • Departments were asked individually what they needed • Lists of specifications were made and users identify what was needed • Develops and user groups • Use hotel facilities, training and development session were carried out • Choose two key members from each department to represent the users of the department • One to one sessions • Developers taking control • Develop initial project team which includes developers and users and complete the project • Developers watch the users over a period of time • Less experienced user answers a checklist of what was required • Higher levels mapped out the system with group meetings

PRESENT INVOLVEMENT		
No.	**Question**	**Response data**
2	Percentage of the total users in the company who were involved in the most recent development process?	Average of 34.4% of total users
3	How much interaction was there between yourself and the development team?	• 3 months-1 year weekly interaction for two months • Twice per week for 5/6 months • One week – half day seminars • 5 full days on specifications • Once per week for 2 hours • A few meetings • 4-5 months with weekly meetings

The responses listed in table 4.7 suggests that even though there are some factors or methods being used there still seems to be some companies who do not see the user as a vital part of the process. If we look at the responses from the question, "how did the process take place?" the responses suggest that users and developers should be interacting well. However, for the question, "what percentage of the users took part in the development process?" the response was an average of 34.4% of the total users of the systems were involved. This suggests that around two third of the users of the systems did not have any say in the development process.

The amount of time that developers and users spent interacting is very difficult to measure due to users' lack of knowledge of project sizes. Users did not really know how much time was spent developing the system. This is quite an important factor, as users need to feel that their part is important in order for them to show higher levels of enthusiasm. This would require further research on the subject of time in relation to user involvement and project size. However, this data shows how much consideration the development team places on the users. For example if we look at one statement 'a few meetings' it is clear that the user was not very involved. On the other hand the statement 'twice per week for 5/6 months' shows a greater level of enthusiasm.

The results of the level of involvement shown in table 4.7 have played a part in causing the problems that are being faced by the companies. Therefore, an increase in the level of involvement may offer the chance for more users to add their views on change and offer their knowledge about the current systems.

Cluster 7: Suggestions for improvement [Users]

Table 4.8: Table showing the users views on the process of the development of information systems

SUGGESTIONS FOR IMPROVEMENT		
No.	**Question**	**Response data**
1	How do you think that the methods that were used could be improved?	• Improve the developers knowledge of the manufacturing processes • System should be reviewed on a regular basis • System should be able to be changed at later stages in the process • Involve all staff at some stage • More feedback from developers • Involve more people from the bottom end • Improve the support facilities • Have a point to go to when problems arise • Make sure that all users are represented • Use some form of questionnaire to generate opinions and ideas • Identify more focused goals and direction so that users are more knowledgeable about the aims of the process • More commitment from higher levels

SUGGESTIONS FOR IMPROVEMENT		
No.	**Question**	**Response data**
2	What factors would help you to be more enthusiastic about using the system?	• Developers keeping their promises • More explanations about what is going on and what is going to happen • Knowing that you have had an input in the system • More feedback • More support • Right functionality to assist the daily workflows • Reliable system • Information being more readily available • Easy to use • Getting information when and how it is required • More involvement • Simplicity • Knowing that improvements have been made • Easier access to information • Training
3	How would you like the system to be developed, e.g. Group, individual consultations etc.	• 100% Opted for using group sessions • Use of brainstorming sessions • Getting managers in on the discussions • Try to appoint a representative • Use small groups • Well structured groups

Table 4.8 shows results from the questions that give the users' opinion of how the process of capturing user requirements can be improved. It also lists the responses to questions that relate to what users feel would encourage them to use the systems that have been implemented.

The responses that have been listed in table 4.8, will be considered when the proposed method is developed. These are individual views that users who were involved felt were important. Therefore, the proposed methods should try to incorporate the responses that fit in with the objectives of the proposed method. One of the most significant responses in table 4.8 is that 100% of the respondents opted for using a group for the interaction process. This indicates that users are more comfortable when they are working in group situations. The improvements suggested by users also relate to increasing involvement, thinking about future improvements and functionality.

4.3.2 Developers perspective

Figure 4.4: Cluster titles for information relating to development from a developers perspective

CLUSTER 8: METHODS CURRENTLY BEING USED
CLUSTER 9: PROBLEMS WITH CURRENT METHOD
CLUSTER 10: SUGGESTIONS FOR IMPROVEMENT
CLUSTER 11: TYPES OF USERS INVOLVED

Figure 4.4 shows the cluster titles that will help to develop the proposed method from the information received from the developers.

Cluster 8: Methods currently being used [Developers]

Table 4.9 shows the methods that the companies have adopted to carry out the requirements capturing process.

Table 4.9: Table showing the methods that are used to capture user requirements

METHODS CURRENTLY BEING USED		
No.	**Question**	**Response data**
1	How are user requirements identified?	• Feedback between users and developer mainly identifies the needs. • Separate the system into categories and assign experienced users to the process • Identify at least one user from each department • Start with expectations and discuss until final specifications are agreed • Developer receives a document with what users want • Carrying out a full study of the system • Identify a need – Create a problems definition document • Users say they need a change and developers help them to change the system accordingly • Senior manager of the company identifies a need for change – Identify what the aims are – explore the business requirements • Team consisting of key users identifies gaps and develops the system. They will also sign off requirements. They work alongside consultants and the supplier of IT packages, e.g. SAP, Baan, IBM, Oracle etc.

METHODS CURRENTLY BEING USED		
No.	**Question**	**Response data**
2	What methods are used to acquire user requirements?	• No real structured method • Interviews • Group sessions • Data flow diagrams • SSADM • Brainstorming sessions • One to One meetings • Flowcharts • Telephone calls • Yourdon • Workshops with business experts
3	How is the analysis carried out?	• Use SSADM • Use entity relationship diagrams • Going through the requirements backwards

After reviewing the responses from table 4.9, the developers have been able to contribute to the development of the proposed method by suggesting some ideas relating to how the systems development process should involve the users. However, these methods have not been identified by the users in such detail. This is one of the major problems with these processes. The users are not fully aware of what is happening.

The methods that are used have advantages. For example, both brainstorming sessions and flow diagrams both help the user to get involved and help to represent the information collected. However, the main problem is the structure that is placed on the use of these methods. In most of the cases, these methods are mentioned because they seem to be good ideas. The problem is the lack of repeatability of the process. They do not suggest a method that starts from the beginning and structures the process until the end of the requirements capturing process. Therefore, the proposed method would incorporate some of the ideas, but to place them in some order so that the process can be carried out the same way again.

The responses from the question " how is the analysis carried out?" shows that there is a lack of knowledge about the analysis of the information received from the users. SSADM has been used which is very good for graphically representing the data. This technique shows flows and relationships that are very important to the development process. It is important for the information to be analysed in order to go to the next stage, which is development. If there is not a thorough analysis, duplication and wrong flows could be entered into the system making it less useful to the users.

Cluster 9: Problems with current methods [Developers]

Table 4.10: Problems that the companies experience that relate to user involvement and systems development

PROBLEMS WITH CURRENT METHODS		
No.	**Question**	**Response data**
1	What are the main problems in the requirements capturing process?	• Time factors • Resistance to technology • Missing requirements • Not enough interaction with users • Users not happy with requirements placed in the system • Lack of understanding between developers and users • Users do not know what they need • The right users are not spoken to • Developers are to far away from the system after the implementation • Users need to be able to communicate what they want easier • Users tend to be impractical • Lack of trust

PROBLEMS WITH CURRENT METHODS			
No.	**Question**	**Response data**	
2	Do you always get the right information after the first interview	0% Yes – 100% No	
3	How long are the systems expected to last	Under 5 years	16.67%
		Over 5 years	44.47%
		When need arises	22.22%
		No time specified	22.22%

Table 4.10 identified the problems that are being faced by the company in relation to user involvement and information systems development.

According to the results in table 4.10, 100% of the developers state that they do not get the correct information after the first interview. This was investigated further by asking the question "what are the main problems in the requirements capturing process?" The results suggest that there is a breakdown in communication between developers and users. This suggests that curing this problem of communication may result in a positive step towards systems success.

Developers have suggested that the time factor is a problem (see table 4.10). This may be caused by the lack of importance that has been placed on this process. Another step towards achieving successful user involvement would be to have a structure that can be repeated. This should help users to get to know the process and less time should be used explaining what is going to happen. Problems may also exist because systems are expected to last such a long time. Today's IT industry requirements are changing very rapidly and therefore a maintenance strategy should be formulated which involves regular checks on requirements. A very high percentage did not really look at this factor and gave answers that show that they expect the system to last as long as possible.

Cluster 10: Suggestions for improvement [Developers]

By interviewing the developers, it was possible to find out how they thought that the process could be improved. Table 4.11 lists the suggestions that were given by the developers.

Table 4.11: Table showing the methods that developers think could improve the process of developing information systems

SUGGESTIONS FOR IMPROVEMENT		
No.	**Question**	**Response data**
1	How would you improve this method of development?	• More user participation • Improve the speed of development • Get users away from working environment • Need guidelines • Increase time given to complete the process • More interaction after the implementation • Stricter documentation • Use a structured method e.g. SSADM • More publication of findings at various stages • Pre issue questionnaires • Encourage more teamwork • More resources

Table 4.11 shows the responses to the question "how would you improve the current methods used?" these are very similar to the users. One of the responses was that user participation should be increased. The other responses relate to structure, time and resources.

The results shown in table 4.11 suggest that the general trend is not to specify a time for the requirements capturing process. Those that have given a time will need to evaluate how much time should be spent on particular projects according to its size and the resources that are available. Those who said that they would like to use the least amount of time possible are taking a risk of missing requirements due to rushing through the process. However, every process should take the least amount of time possible that achieves the goal, but the process should not be rushed just to save time.

Cluster 11: Types of users involved [Developers]

Table 4.12 lists the types of users that the developers produce systems for. Users need to understand the process of capturing user requirements, if the level of complexity is too high then the users could lose interest and spend more time trying to understand the process. This will waste valuable development time.

Table 4.12: Table identifying the types of users involved in developing systems in the companies who participated in the survey

TYPES OF USER INVOLVED			
No.	**Question**	**Response data**	
1	Who are the users of the systems that are produced by the systems department?	High level of experience	5.56%
		Medium level experience	5.56%
		Low level of experience	16.67%
		Mixed experience	72.22%
2	How much technical knowledge of the development process should users have?	High	5.56%
		Medium	66.67%
		Low	27.78%

The results gained from table 4.12 suggest that there is a wide range of abilities within the organisations used in the survey. The expectations from the development teams are not high as only 5.56% expect a high level of knowledge of the development process (see table 4.12). These results suggest that users have the ability to take part in the process at a level that they can understand. The conclusion drawn is that the expectation should not be too high. This means that the methods used to interact with users should be as easy to understand as possible.

4.4 Findings

This chapter used clusters to group questions together so that conclusions can be drawn regarding specific aspects of user involvement. The evidence suggests that there is a need for the research, and that the users and developers are able to contribute to the development of a new method of involving users in the development of information systems.

According to the findings of this chapter, the evidence received from the developers and users suggest that user involvement is an important factors in developing information systems. An increase in this involvement is likely to lead to more efficient information systems once the correct methods are used and also improved usage of the systems.

Within the survey some users found alternative manual methods of finding information. They have stated that the systems are not giving the right information or they find it hard to use the system the way that they have been told to. There are problems with functionality, as users cannot find what they want when they want it. They state that the functionality is there but they don't know where it is. This has lead to them having less faith in the system and the development process.

One factor that the study has found is that the users are not being involved as much as they would like. The users have expressed that they feel that they would like to know more about what is happening and would like more feedback. Therefore, they have not been able to input the relevant information into the specifications. The overall feeling is that users are not happy with the systems being used. They would like more interaction. However, they are still using the system. This shows that they will use the system as much as they can until it is improved.

The time element is dependent on the size of the system and the size of the project, as the larger the project, the more time that should be reserved for the process of capturing user requirements to be completed effectively. However, it is important that each development team considers this factor.

The factors relating to the technical aspects of the method are not really considered by the users. This is mainly due to a lack of knowledge that they have about systems development. However, users seem to be quite positive towards the use of groups within the process. This will encourage changes to be made collectively rather than there being too much individual pressure.

The results from the survey show that the developers are very conscious of what can produce an efficient system. There are quite a few methods or skills that can be adopted within a new methodology. The ideas that have been generated show that users are not happy with the way that information is collected. There would definitely be a need for a more structured approach and acceptance of the need for more user involvement from developers, which has been achieved.

After reviewing the discussions that were based on the findings in this chapter it is also suggested that users and developers are generally dissatisfied with the levels of interaction that has been taking place. However, the users in many cases are content due to their limited expectations of what could happen. This conclusion is based on the fact that users expressed a concern about the levels of interaction. The users also suggested many problems with the current systems, but they show an average level of satisfaction, rather than a low level.

Time has been a concern according to the findings. This is a factor that is dependent on resources. The results in this chapter suggest that the time allocated to the process of capturing user requirements needs further investigation by each company. This should produce an optimum time that would suit different project sizes, according to the types of users, the complexity of the systems and the total time allocated to the whole project.

The methods that are currently being used have proven to be unsuccessful. This conclusion is based on the problems that are currently being faced. However, the developers and users have been able to suggest a collection of ideas that could improve the way that the process is carried out.

The results obtained in this chapter also suggest that the developers are aware of what can produce an efficient system. The methods that have been mentioned include the use of structured methods including SSADM. However, due to the resources available developers have used an ad hoc approach. This is one of the reasons why the results of this chapter have shown a negative conclusion towards the current systems. The major problem with an ad hoc approach is that it is difficult to repeat. This is not beneficial to users as they cannot get used to one method. This has also led to problems with the functionality that the systems provide. A lack of functionality leads to the system not being used as it is supposed to be used , which may lead to incorrect outputs.

Developers have also suggested that the interaction between the users and developer needs to be improved. This is a positive step towards an increase in user involvement. Therefore, once the developers and users have the right level and type of interaction there is likely to be an increase in usage of the systems.

The objectives of the survey of users and developers seems to have been met. The results that relate to objective 1 suggest that user involvement during systems development has not been considered in great detail within the companies visited. However, the general view is that an increase in the involvement of users is required. This is shown by the high percentage (76.47%) of users stating that they would like the level of involvement to be increased. The results also show that there should be an average of a 51% increase in the involvement of users. The results to meet objective 2 show that the companies visited do not have high usage of structured methods. However, methods such as SSADM and YSM were mentioned.

CHAPTER 5

THE DEVELOPMENT OF URCAM (USER-ORIENTED REQUIREMENTS CAPTURING METHOD)

5.0 Introduction

Avison and Fitzgerald (1995) state that in the early days of information systems development, the emphasis of computer applications was towards programming, and the skills of programmers were particularly appreciated. This meant that the systems developers were therefore technically trained but were not necessarily good communicators. Avison and Fitzgerald (1995) also state that this often meant that the needs of the users in the application area were not well established, with the consequence that the information system design was something inappropriate for the application.

A number of methodologies have been developed in the past that offer the user of the methodology a structured approach to developing systems. They include methods such a SSADM, YSM, OOA etc. However, the literature review suggests that there is a need for a new approach. Purvis and Sambamurthy (1997) state that, spurred by the search for prospects of enhancing user participation, researchers and practitioners have evolved and enhanced the variety of development methodologies. They also state that a crucial element of the traditional IS design methodology is that it limits user participation to a consultative role, where the bulk of the decisions are made by the IS developer or analyst. Therefore, this chapter aims to address this issue and develop a method that assists the interactive process between developers and users, offering the user a chance to play a more important role.

5.1 Generation of the factors to consider

After reviewing the previous literature together with interviews within industry, a broad set of factors have been gathered regarding the requirements of a new method. These factors are shown in tables 5.1-5.3.

Monk (1991) states that the specification of users' requirements is traditionally seen as the first step in the development of new information systems and products. He states that it is the initial part of the process of turning the demands (requirements) of the users or clients into technical specifications for systems design and development. Therefore, understanding should be the first criteria for judging specifications. These descriptions also point in the same direction as that of Demarco (1979) who states that the success of the specification process depends on the product (the target document) being able to serve as a model of the new system. To the extent that it helps you visualise the new system.

Table 2.5 showed a comparison of the systems development methods that have been reviewed within this thesis. The most distinctive result is within the User / Developer interactive process category. The results in this section showed that the methods are not clearly considering this area of systems development. DSDM, RADD, RAD and JAD are attempting to increase the involvement of the users by encouraging groups of users and the developers to occupy the same room, in order to carry out the development process. This suggests that there are methods that see the importance of the user.

The results from table 2.4 led to a further investigation to identify whether there were any methods that claim to have been developed to consider the interactive process between the developer and the user. A selection of these methods were found and reviewed.

Throughout the literature review, the systems development and user requirements capturing methods have shown a number of strengths and weaknesses which have assisted the study in mapping out the route towards the specification for the new method. Figure 5.1 illustrates the points, which are considered as being the strengths to help achieve the desired goal.

The survey suggests that a group situation would encourage users to feel more involved. The literature has been able to identify methods that encourage teamwork to be implemented in the development process. This is how the need is satisfied. Therefore, the literature would be discussed in more detail to help to solve the problems of the survey.

If we take a look at the literature we can see that there are many methods that have been used in the past. The relevant factors can be looked at in more detail in table 5.1. The clusters in the survey of users and developers present lists of the factors that should be considered from an industrial perspective.

Table 5.1: Factors from the literature and their relevance to the development of URCAM

Factor	Relevance to the study
MoSCoW Rules.	These help the developer to characterise requirements and also helps to justify to users why specific requirements are added or deleted from the specification.
Active User Involvement.	First principle of DSDM. It aims to improve the interaction with the user.
Teamwork.	The encouragement of people working together to achieve a common goal.
Revise Changes.	Once a change has been made to the system they are checked by users to ensure the validity of the information.
Integrate testing throughout the lifecycle.	This helps to reduce the time wasting in the development process, ensuring that the users do not have to be interviewed too late in the process leading to drastic changes in the system specification.
Workshop used to capture requirements.	This method brings the users and developers together in one area that should save time and resources. The users can run through processes with other departments or representatives of systems so that specifications can be agreed while they are all together.
Technology.	Tools can be used to graphically represent the data that is received from the user while the process is being carried out. This should also save time and produce results that users can see and hopefully understand.

Factor	Relevance to the study
Higher level of involvement.	Users are supposed to be present in the workshop and not just high level employees who can make the system look as they want it to look.
Breaks the process into well defined sections / views.	Here each area of the system can be looked at individually or sections can be looked at that relate to each other.
Use of different levels.	The system can be looked at from different levels, this allows the developer to look back at areas and find mistakes where users are not receiving information that they are supposed to receive.
Ability to change according to varied circumstances within the company.	The systems development team must be able to use the information that they receive and see how the information being transferred around the system is used. This should then prompt them to be able to change the layout of the system and ensure that users are obtaining information that ensures that the systems works in the best interests of the company.
Breaks up phases.	This idea helps to structure the whole process from beginning to the end.
Graphical representation of data.	Different symbols are used in this method so that different diagram properties can be distinguished from each other.
Uses timescales within the representation.	The diagrams used show how the different operations are effected by time.

Factor	Relevance to the study
Nets show the difference between process and decision activities.	Here the good use of diagrams helps to distinguish between areas where decisions have to be made. This helps the person looking at the diagram tell what decisions have to be made at various parts of the system.
Breaks down system into layers.	This diagrammatic technique allows onlookers to see how different levels are connected.
Phases involved help to define the 'As Is' and the 'Should Be' in a well-defined manner.	This can structure the process making it clearer to the developer what is expected within this area of the development process.
Shows links between specific areas of the system in a well-defined manner.	This is carried out by using layers that look at the subject and then structures that subject. The subject is a rough-cut or user view of the system being investigated and the structure is a technical diagram representing the subject.
Objects are well defined.	Each object has a list of attributes that help to define it. This helps when the system is looked at, at a later stage.
Extracts the main features to be examined.	This can be useful if a specific area of the system needs to be looked at in more detail.
Models involved encourage cost-effective development.	This method identifies the concerns on the system, which highlight areas where cost-effectiveness can improve the system specification.
Diagrammatic representation.	Good for presenting information to people who can understand the diagrams.
Different symbols representing different aspects of the system.	Developers can see what is happening between different entities within the system.

Factor	Relevance to the study
Used extensively in industry.	Reputation gives it some validity.
Breaks down to show different levels of detail.	This enables the developer to show the diagrams at different levels to different interest groups.
Very structured approach.	Rules are set which assist the developer to create diagrams that represent different aspects of the system.

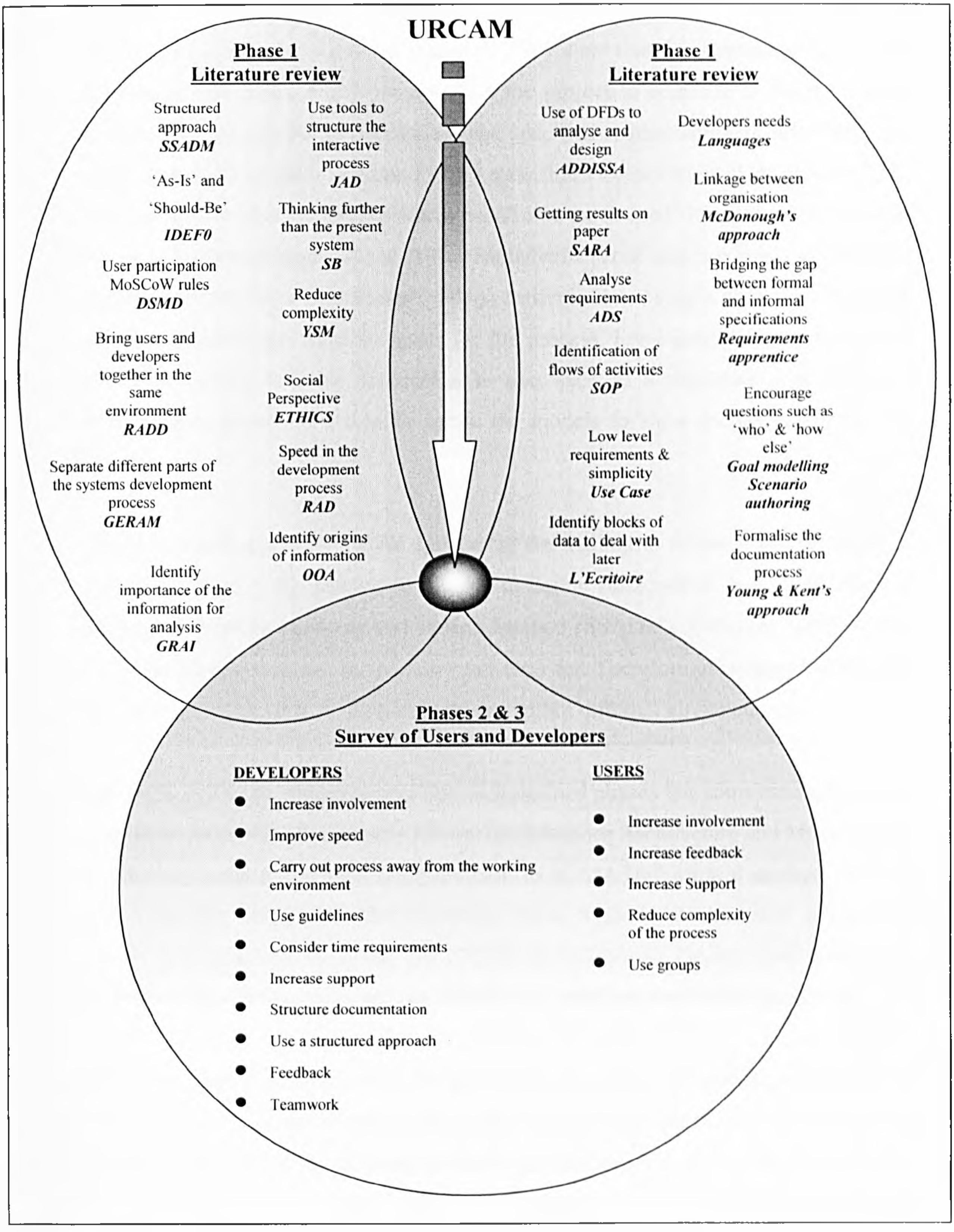
Table 5.2: Factors to consider from the literature

FACTORS TO CONSIDER
MOSCOW rules
Active user involvement
Teamwork
Revision of changes
Integration of testing throughout the lifecycle
Use of workshops
Use of technology
Breaking down process into well defined sections/views
Use of different levels
Change according to company circumstances
Break up process into phases
Use of graphical representation
Use of timescales
Representation showing areas where decisions can be made
Distinguish between different elements of the system
Phases help define as-is and should-be
Show linkages between different areas of the system
Use of well defined objects
Main features can be extracted
Diagrams showing area of concern
Use of symbols to show different elements in the system
Use of a structured approach

Table 5.3: Factors to consider from the survey of users and developers

FACTORS TO CONSIDER
More interaction is necessary
Involvement does make a difference
Systems are old and slow
System creates errors
Functionality is not found
Not user friendly
Users find ways around the system
System does not do what users would like it to do
System needs to be reviewed at specific times
Make sure that users are represented in development meetings
Involve all staff at some stage
Improve the support facilities
Increase commitment at higher levels
Use user group sessions
Try to appoint a representative for the user group
Improve the structure for the process
Ensure user input is recognised

Figure 5.1: Strengths from existing methods and the survey of developers and users used to achieve the desired goal



5.2 Discussion of the factors to consider

Folkes et al (1992) state that traditional methods did not satisfy business needs. The answer was to develop a phased approach. They state that this approach would offer options at each stage, which would allow the project to continue to the next stage, abandon the project, backtrack and change specified requirements, or alter decisions made earlier. Coad and Yourdon (1991) state that Object Oriented Analysis (OOA) consists of five major activities which need to be performed. They should be seen as stages or sequential steps. Komi (1995) identified a set of stages within the Graphical Results and Activities Interrelated (GRAI) method. These stages give some direction to help the developer form the goals for the process. For example, they identify the groups necessary and the approaches to use, such as a top-down approach or a bottom-up approach, and then to create the models to show inconsistencies in the system.

This structured approach enables the user of the method to follow a pattern or set of stages. Structuring the process also helps to define rules, which is a main factor of Structured Systems Analysis and Design Method (SSADM). Different systems have many differences that one method may not cater for. Therefore, it is suggested that the new method be flexible in relation to its use within different circumstances.

The breaking down of the process into well-defined phases has some similarity to the structured approach. The Generic Enterprise Reference Architecture and Methodology (GERAM) method and other methods such as ICAM Definition 0 method (IDEF0) and Computer Integrated Manufacturing Open Systems Architecture (CIMOSA) adopts this approach. However, the IDEF0 method breaks the structure down even further to identify the As–Is and the Should–Be considerations within the process.

The survey has identified a need for an increase in user involvement (see chapter 4). This is a factor that has been included in the Dynamic Systems Development Method (DSDM). The aim of this is to ensure that users get involved to a greater extent. This is a factor that the survey has suggested is important. Systems developers and users recognise the fact that to produce an efficient system users need to be involved effectively. The results from the survey indicate that 94.12% of the surveyed users said that their involvement affected their willingness to use the newly developed system.

Teamwork represents a common factor among the Rapid Application Development (RAD) techniques and DSDM. The expectations require effective user involvement and effective teamwork to ensure that correct information is given to the developers. It must also be noted that the team should also consist of the developer. This should encourage the responsibility to be spread so that everyone involved will be able to play an active role. The aim is to bring users and developers together and form teams that would be responsible for developing specified parts of the system.

Stapleton (1997) states that a typical DSDM project will have teams. This is a very useful factor, as it should save company time and resources. It would mean that the group is together and should produce more beneficial answers to questions asked simply because they can help each other or back up information that is given.

The use of teamwork relates very closely to the use of workshops and technology. This is one of the RAD techniques where users and developers are placed in the same room where the specification takes place. However, the technological side, which is suggested in the Joint Application Development (JAD) area of this approach is something that may not prove to be so beneficial to the user as it is likely to be something that they are not too familiar with. This may cause problems with understanding and acceptance of the method.

Graphical representation is very important for portraying the system to the users as it shows a representation of the process. The aim of the representation of the data to the users is to let them see how data flows around the system according to their wishes. This means that they must be able to understand this representation. There are a few methods that encourage this type of representation for the users and developers. However, the same representation is used for the developer. Wainwright (1996) looked at the identification of significant entities and the understanding and interaction among them. It may then be possible to assume that the users are expected to have the same level of knowledge as the developers when looking at these diagrams. OOA, SSADM, IDEF0 and GRAI all have very precise graphical tools which are used to show data flows.

A simplified version may be more beneficial to the user. However, the main advantage of these graphical tools is that developers can represent the system in a way that will bring out the different elements within the system. SSADM uses different symbols that are defined and set so that the developer may refer back to them at any time and still have a thorough understanding of what is happening within the system.

The survey of users and developers suggests that feedback is a factor that can affect the users in the development process. Roche (1998) and Martin (1991) have looked at this element within the RAD framework and have identified that speed is an important element. However, the speed should not be used to compromise the quality of the system that is developed. This suggests that in order to achieve the best quality there must be adequate testing and feedback so that the correct information is input into the system.

Feedback relates to the revision of changes that are being made to the system. This factor of feedback was very important within the survey as the users felt that they needed to know more about what was happening throughout the whole cycle. Once changes are made, DSDM requires there to be some form of revision and testing which should then be agreed with the user, so that the process can continue.

Change is highly likely within this development process and decisions have to be made which would initiate change. These decisions should be made in line with the companies goals and objectives which have been suggested within the CIMOSA method. However, the users must be convinced that there are reasons for specific changes. For example, if a user has requested a function to be added, how does the developer justify not adding the function to the new system? The MoSCoW (Must have, Should have, Could have, Would have) rules give requirements categories which help to show why requirements are kept or disregarded. This is given by the DSDM approach to systems development. It may be something that users could accept.

Yu (1998), Rolland (1998), Potts (1997), Plihon (1998) and the DSDM consortium (1997) have all cited work on goal modelling and scenario authoring. The main strength being that the method encourages users to ask "who" and "how else" questions. This is very similar to Structured Brainstorming, which encourages users to think further than what is currently happening. This also relates to the 'should -be' consideration within IDEF0. This suggests that these ideas can assist the user to think about further possibilities that could improve the way that their work is currently being completed.

Reubenstein (1991) reviewed the requirements apprentice and identified the importance of bridging the gap between informal and formal specification. This suggests that the level required at the user interactive stage must be able to be understood by developers. Therefore, both developers and users need to be involved in the evaluation of the new method, which will be carried out later in the thesis.

ADDISSA uses DFDs to analyse and design systems. This is very similar to SSADM, which uses the same principle. Shoval (1998) and Babin (1991) reviewed this method and suggest that it is a refinement of structured analysis. This is also similar to SOP (IBM, 1961) and ADS (Lynch, 1969). These also encourage the analysis of the requirements together with the identification of flows of activities within the system. Once this identification of the flows within the system have been identified, the developer can then describe the linkages between the organisational factors of the system. This is described in McDonoughs approach (McDonough, 1969).

Another weakness of the systems development methods have been the lack of importance that has been placed on the formalisation of the documentation process. Young and Kent (1958) developed an approach to encourage this documentation process. The aim is to have a final document containing the user requirements. This suggests that these documents may be required for future reference. This shows some similarities to SARA. Lor and Berry (1991) reviewed SARA and also identified the importance of getting requirements on paper.

The later stages of the requirements capturing process are described by Zave (1991), Teichrow and Hershey (1977), Alford (1977) and Fischer and Walker (1979). They have cited work on languages such as PAISLey, PSL, RSL and SVL for the high level design stage. This is further than the aim of the study. However, it is considered to identify the relationship between the capturing process and the final development of the Information System.

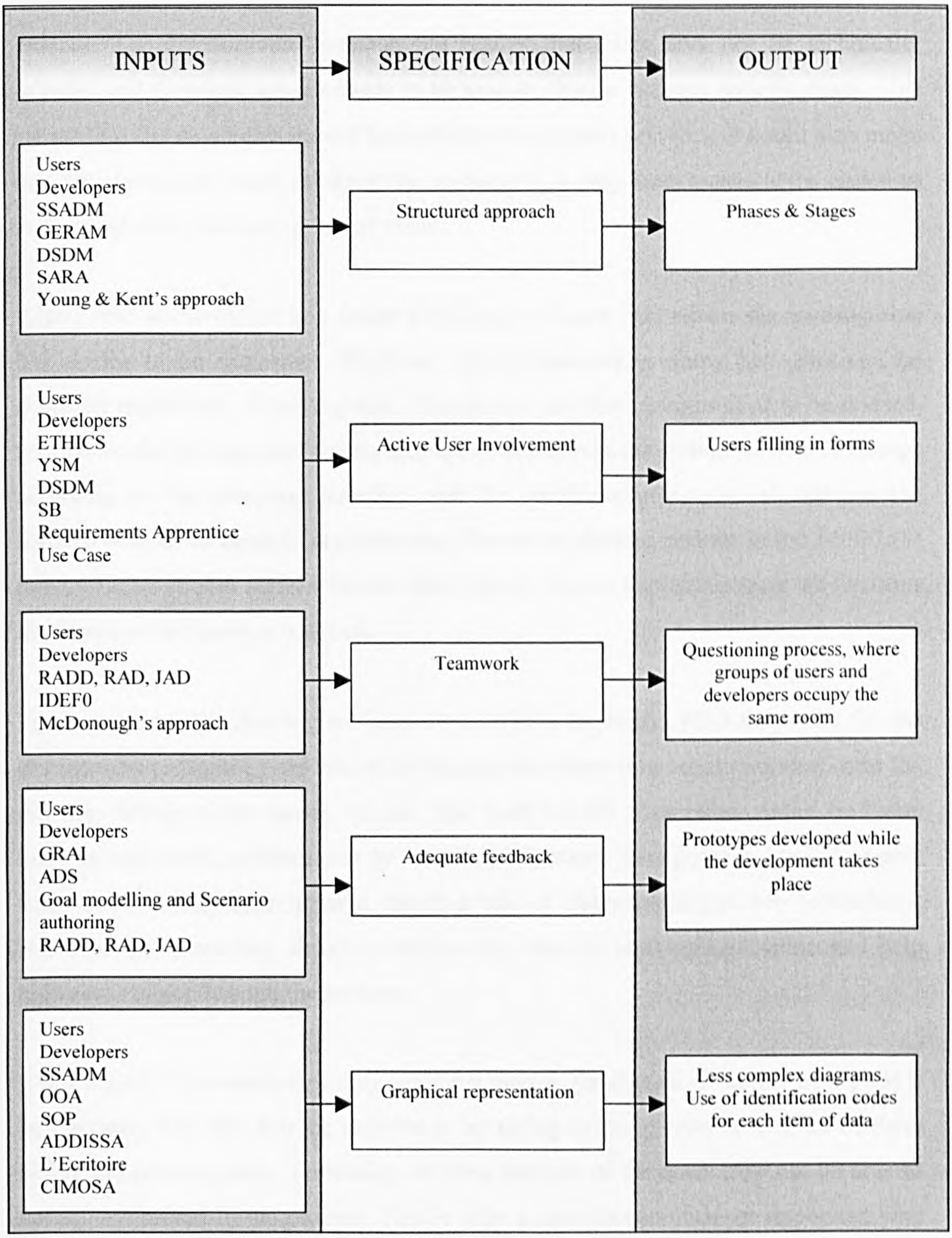
5.3 URCAM development specification

In order to arrive at a specification for the proposed method the key factors that have been discussed will be broken down into groups that will assist this process.

The strengths and requirements from phases 1, 2 and 3 have been identified but they are still separated into the different areas of the study (literature and survey of users and developers) (see figure 5.1). These will be combined to create a specification for the proposed method.

Figure 5.2 shows how the points from phases 1, 2 and 3 have contributed to the development of the specification and how the specification should lead to outputs which will be included in the new method.

Figure 5.2: Inputs to specification to final output



The survey of users and developers has identified that users have been in situations where they do not understand the process that is being carried out. This is a very important factor as it could lead to users not giving the correct information or them spending too much time trying to understand what is happening. Therefore, less time is being spent collecting information from them. This factor is also important after the process. The development team should realise that users may not be technically oriented and therefore are not likely to be able to change systems specifications. This means that the developer should be available for support services. It could also mean that the developer must produce the system in a way that makes it is easier to understand from the users point of view.

Active user involvement is a factor identified in figure 5.1, which the investigation has shown to be important. However, the methodologies show how this can be achieved rather than achieving this. The factors say that changes need to be revised, testing should be integrated throughout the process, systems should be able to change according to the company direction and the development process should use the MoSCoW rules to classify requirements. The most obvious pointer is the MoSCoW rules, which includes various factors that help developers explain to users why various requirements are used or not used.

Teamwork has also shown great importance within the study. Workshops and the use of computer technology are two of the factors that show how teamwork can assist the process. Within these issues we see that with CASE (Computer Aided Software Engineering) tools, systems can be developed quicker. This point is likely to cause users to be slightly apprehensive, due to a lack of understanding of new technology. However, the workshop situation enables the users to back-up each other and help each other to get through the process.

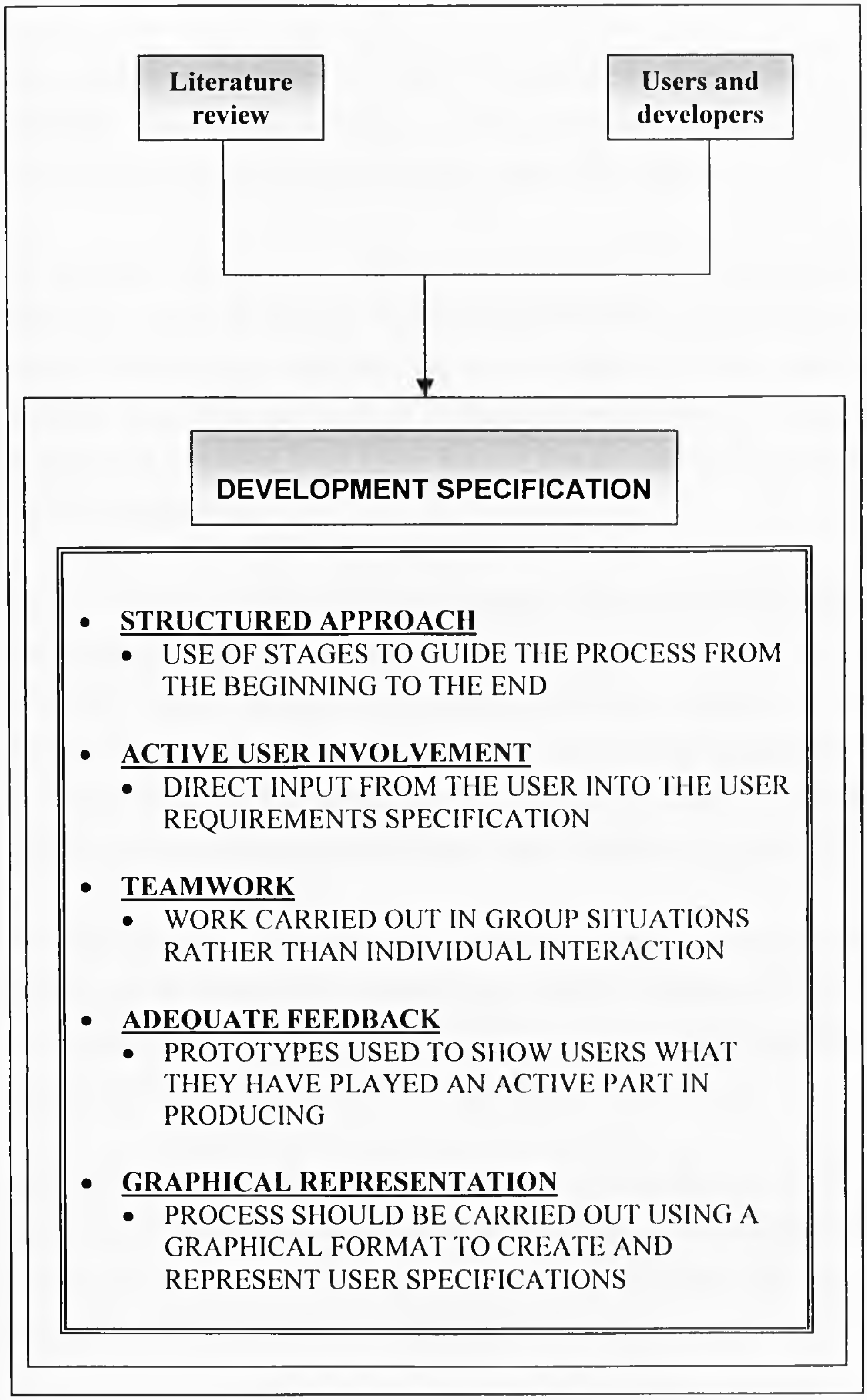
Users need to be consulted or they can rebel against the system no matter how good it is. One way that this may be avoided is by trying to make contact with all of them before the process starts. Following on from that, all of the users may not be able to get fully involved in the process. This is why a representative can be appointed who will be able to feedback information between developers and users.

The user consultation needs to continue throughout the process. This leads to the area of feedback. This area can help the user to feel more important and help the system to look and feel as they would like it to. The user should agree with the end product before it is given to them to use for the final time. The results of this stage being left out are; users complaining about the system not giving them the functionality that they need, systems producing incorrect information and finally, users finding ways around the system.

Methods used to graphically represent the information received from users are very important to the user and the developer. Firstly, the user must be able to understand what is happening. Therefore, they need to be able to understand the process graphically. The diagrams used in the development process must be laid out in a way that is easy to understand. The process should not take them long to understand. If there are boxes to fill in or flow charts to read, these must be as simple as possible so time is not wasted learning instead of developing. Secondly, the developer needs to be able to identify information quickly and easily.

It is now possible to produce a new list of key factors that represent the specification for the new method. This is shown in figure 5.3. It is important to note that the specification must not be used alone, as it does not show the detailed information that is required to carry out the process. However, it does represent the key elements that can be expanded to assist a situation where there is a systems development process that involves users.

Figure 5.3: Specification that will be used for the new method



5.4 URCAM

The specification illustrated in figure 5.3 outlines the foundation of URCAM. URCAM is a method that contains a set of stages that the developer is encouraged to go through. Each stage is described in this section of the thesis.

The first task is to encourage the developer to think about what they are going to achieve as a result of carrying out the process of capturing user requirements. The thoughts should include those that are shown in figure 5.4, but do not have to be limited to those. These are based on the factors shown in table 5.2. This should help the developer to address other issues that are directly relevant to the company that they are working with.

Figure 5.5 shows a diagram of how the interaction can take place. This is based on a RAD workshop (The aim is for a group of users and the developer to be in the same room. The format of the room is left to the discretion of the members involved in the workshop). The room should consist of a table that allows the users and developers to interact comfortably. There should also be a board in the room so that diagrams can be drawn to represent any process that may need to be described graphically.

URCAM encourages the developer to think about the type of users that will be involved in the process, for example their level of experience of carrying out a systems development exercise. This will help them to gear the communication to the right level so that the users will be able to understand the process.

Each activity that the developer should go through with the user is placed within URCAM. The activities are described within each stage of the method. However, the developer is free to adjust the method to suit their own needs. The forms that are required in URCAM are shown in appendix D.

Figure 5.6 shows the preliminary stages of URCAM. These stages will lead the developer towards the use of the URCAM development forms. There is a feedback arrow to suggest that any stage can be completed again if mistakes occur in earlier stages. Figure 5.7 identifies the considerations for the user group. These considerations encourage the developer to be more thorough and ensure that users are kept aware of the process and the final result.

The requirements capturing phases are shown in figure 5.8. These include the stages that are involved in understanding the present system (if there is no current system then the developer can direct the process to the proposed systems phases) and creating the proposed system.

URCAM consists of a set of forms which can be used as templates for capturing information from users. These forms offer a change from the way that development has been carried out in the past. As previous methods seem to depend on verbal communication from users. There is also an ad hoc approach used in industry according to the survey (see chapter 4). URCAM allows the user to fill in forms and a structure is also added to help the process to be repeated.

The URCAM forms are the tool that attempts to give the user a greater role in the development process. These forms are shown in appendix D (together with help sheets). However, an example of how the forms can be used are shown in this section of the thesis.

The first form that needs to be filled in is the reference table. This table will list all of the departments, manual storage areas, systems and positions (people) that are involved in the systems development project. Each of these departments, manual stores, systems and positions will be given a code which is used as a reference throughout the project. This table is shown in figure 5.9. Figure 5.9 also shows form 1, which is used for the users who are involved in the project. The department, manual store, system and position that the user is going to represent is documented in this form. The information should be contained in the reference table. Therefore, in this example the position that the user has is the quality assurance controller, which gives them a reference code of D8.1. They represent the quality assurance department, whose code is D8. This is also filled in for reference purposes. From this point onwards the identification code for the user will be D8.1

The next form (form 2) is used to identify which other departments, manual stores, systems or positions the particular user interacts with. This form is shown in figure 5.10. In this case the identification code is the same as in form 1 (D8.1). The user is required to tick the boxes with the reference codes that represents the departments, manual stores, systems and positions that they interact with (e.g. send or receive data to and from). Figure 5.10 shows that the quality assurance controller (D8.1) interacts with Dl (Tyre 1), D3 (Washington) and S8 (Prium).

Figure 5.11 shows how form 3 can be filled in by the user. Again the identification code is copied from form 1. The user then fills in the codes of the departments, manual stores, systems and positions that they send data to. In this example data is sent to D1 and D3. The form also contains boxes for the user to describe what type of data they send to each particular departments, manual stores, systems and positions.

Form 4 is very similar to form 3. However, rather than asking what data do you send, form 4 asks what data do you receive form other departments, systems etc. The principle for filling in form 4 is the same as for filling in form 3. This is shown in figure 5.12.

Form 5 attempts to document more detailed information from the user about the data that they receive. Figure 5.13 shows this form. For each item of data received there is a box with the data code and an empty box for the user to write down what they do with each item of data that they receive.

Finally, in this example the data flow diagram representing the information collected from the users is drawn. This is shown in figure 5.14.

Figure 5.4: Initial thoughts

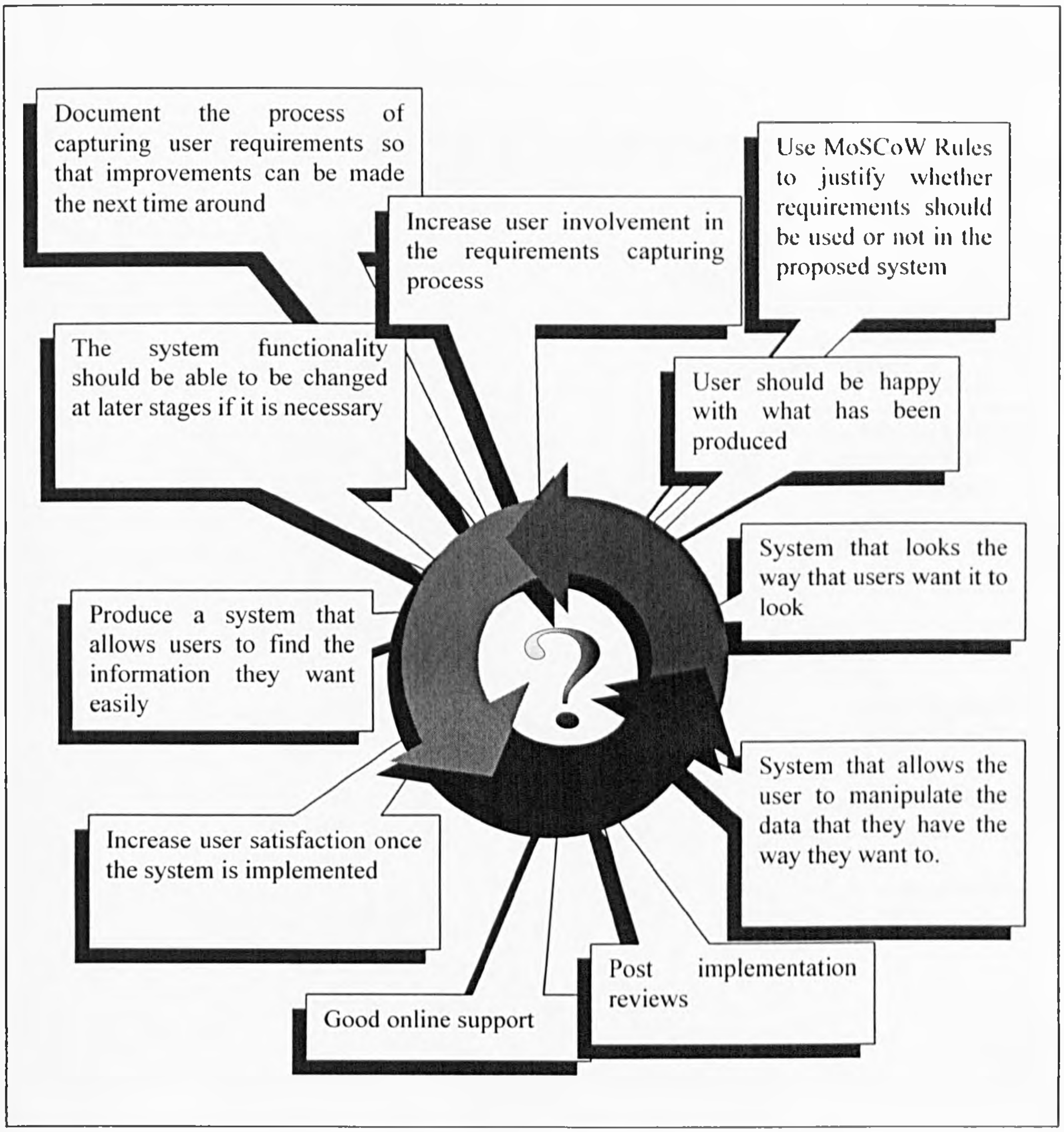


Figure 5.5: Diagram representing a possible format of the room where the interaction with the users and developers can take place

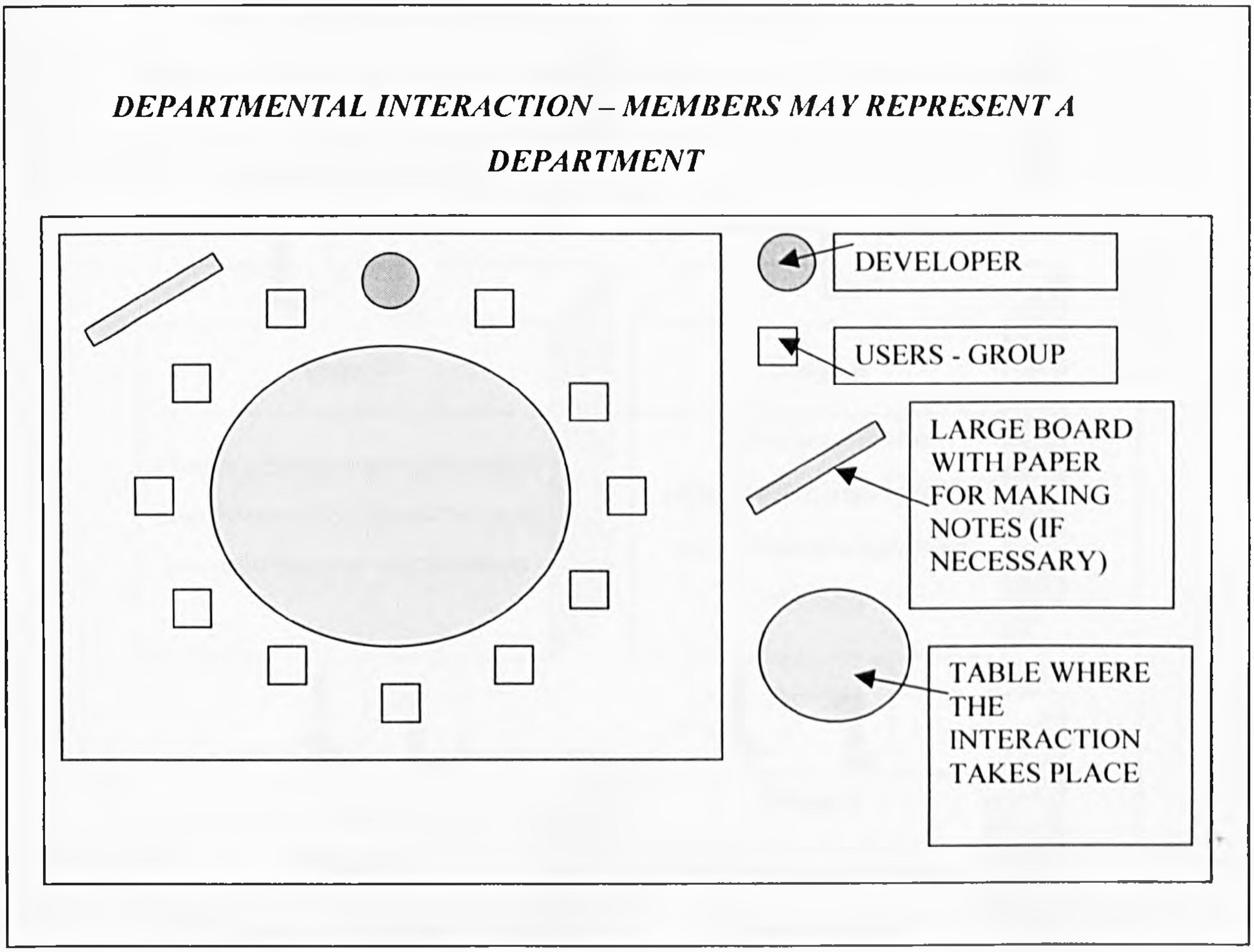
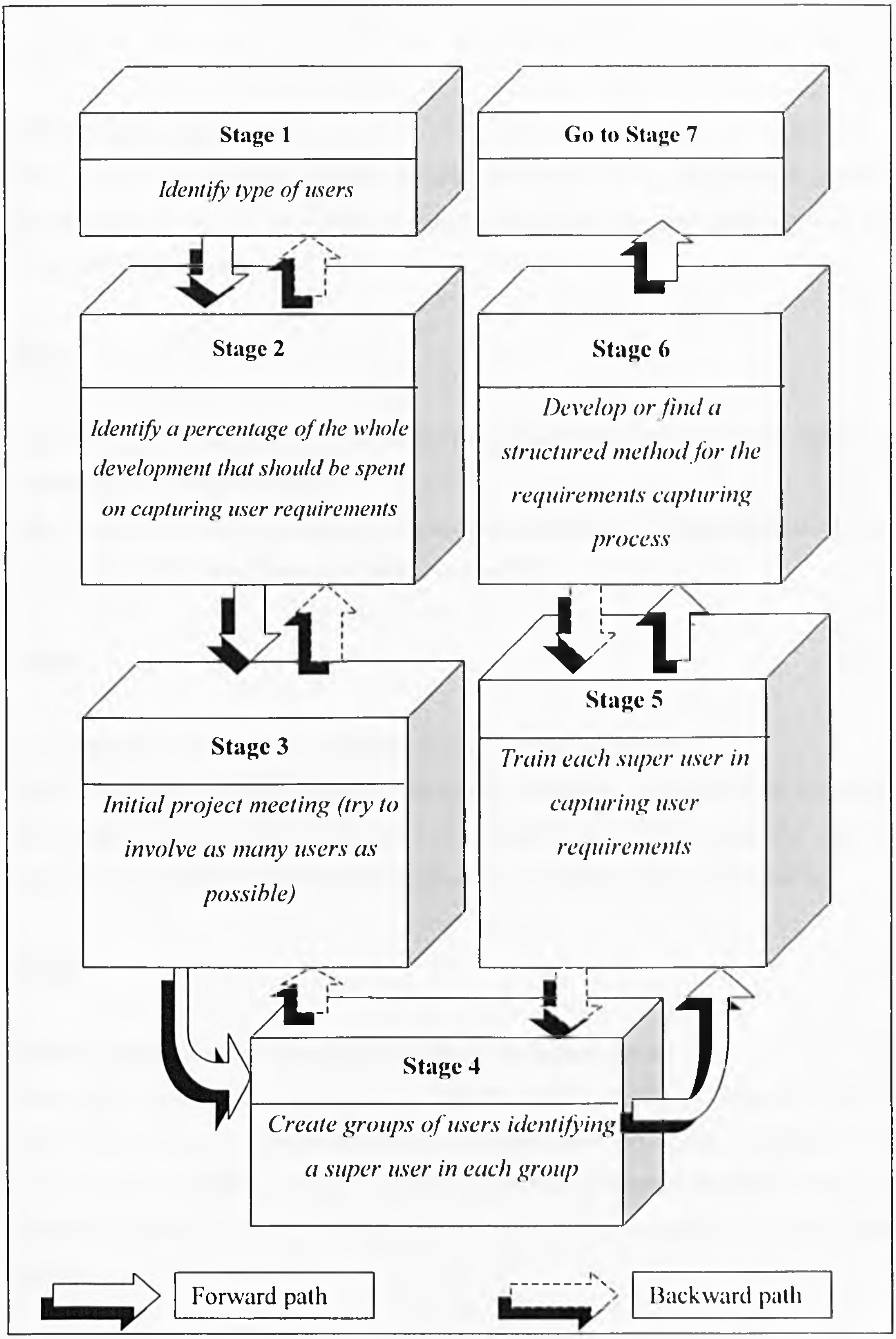


Figure 5.6: Preliminary stages



5.4.1 Preliminary stages

Stage 1

Identify types of users.

This means that the level of user knowledge in relation to information systems development should be identified so that the developer can communicate with the users at the right levels.

Stage 2

Identify a percentage of the whole development time that should be spent on capturing user requirements.

The time spent on the capturing of user requirements should be specified rather than trying to complete the process as quickly as possible

Stage 3

Initial project meeting (try to involve as many users as possible).

This is important, as all of the users will not be available to go to all of the meetings. This stage would ensure that the views of the workforce are heard and that they all agree to some extent with the need to implement or improve the current system.

Stage 4

Create groups of users identifying a super user in each group.

The main reason for using groups is so that the users can prompt each other and so that information can be backed up by the different members of the department. The second aim is to identify a super user who will be able to interact with the developers and users to form a bridge where an understanding can be achieved between both parties.

Stage 5

Train each super user in capturing user requirements.

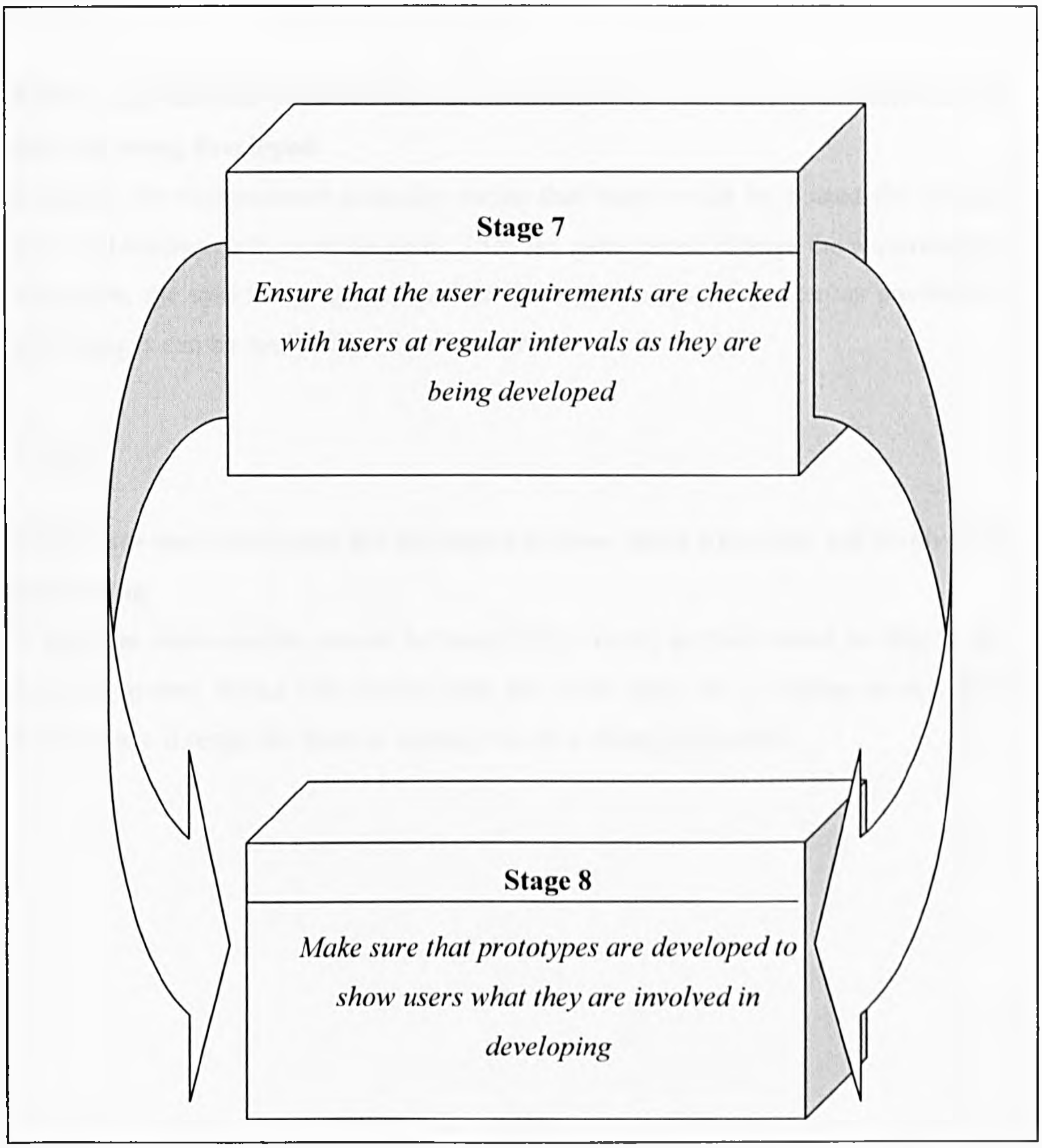
In most cases it is not likely that all of the users will be able to go to the sessions. Therefore, the super user must be responsible for communicating with the other users, which means that they must be able to attend all of the relevant meetings.

Stage 6

Develop or find a structured method for the requirements capturing process (e.g. use the URCAM forms, appendix D).

This would help the process to be repeated more easily and also for different developers to get involved with the projects. A standard structure will also help the users to have a better understanding of the process. This means that they can interact more and feel more aware of the process that is taking place.

Figure 5.7: User group important considerations



5.4.2 User group considerations

<u>Stage 7</u>

Ensure that the user requirements are checked with users at regular intervals as they are being developed.

Checking the requirements normally means that there would be a need for change. This will happen up to a certain point. Then the users cannot change the requirements. Therefore, the systems progress must be shown to the users as often as possible so that changes can be made before it is too late.

<u>Stage 8</u>

Make sure that prototypes are developed to show users what they are involved in developing.

A physical representation would be beneficial to users, as they would be able to see how the system works and maybe even test their usage on a running model. This would make it easier for them to identify where a change is needed.

Figure 5.8: The requirements capturing phases

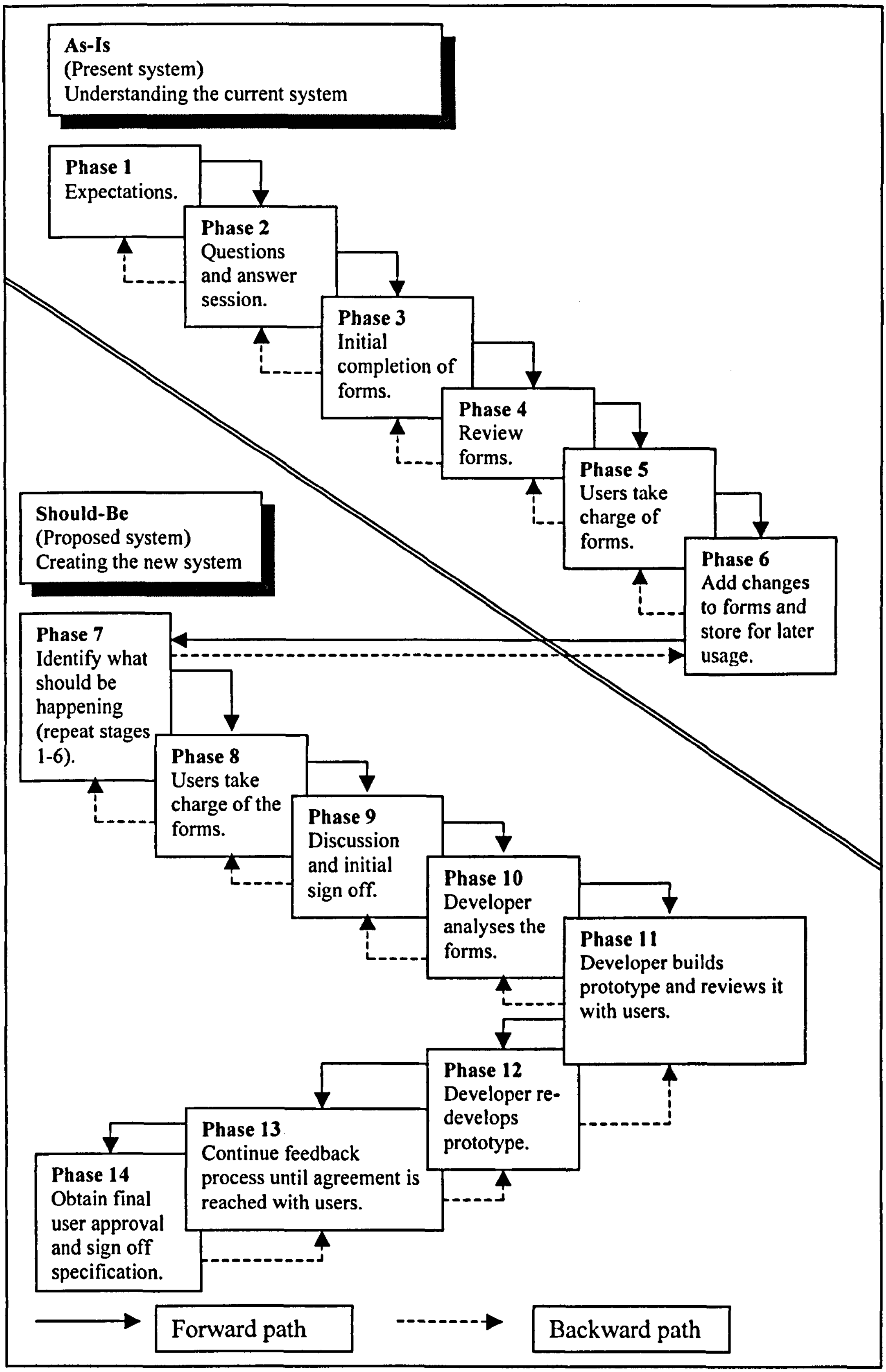


Figure 5.9: Form 1 with reference to the reference table

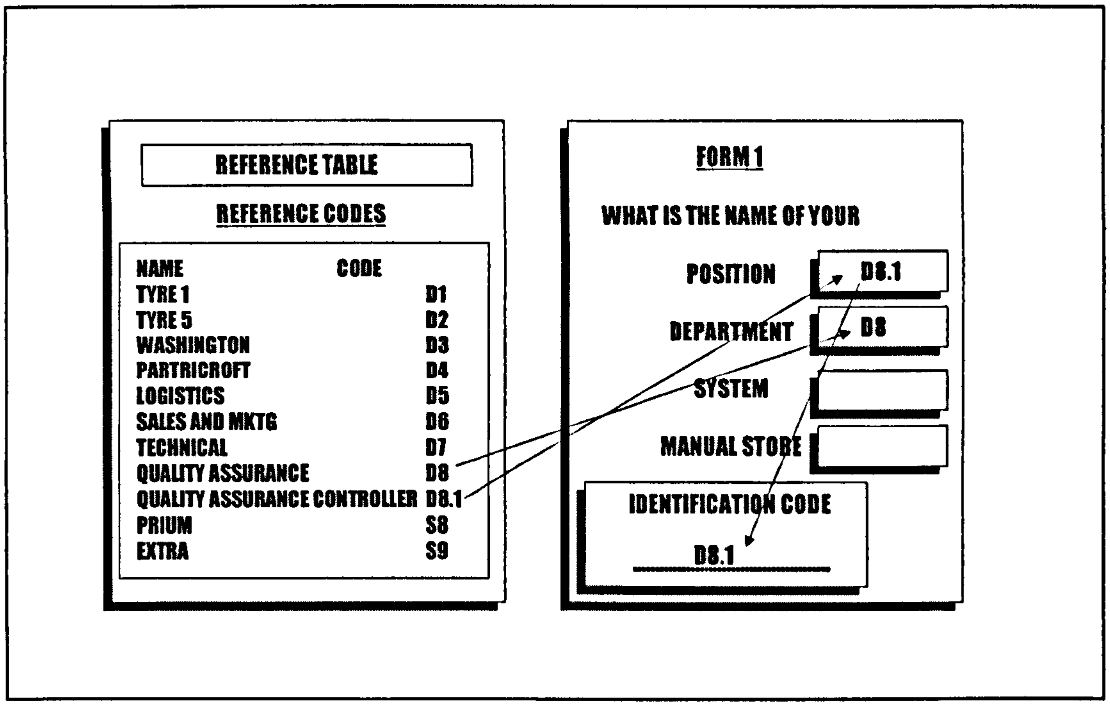


Figure 5.10: Form 2 with reference to form 1 and the reference table

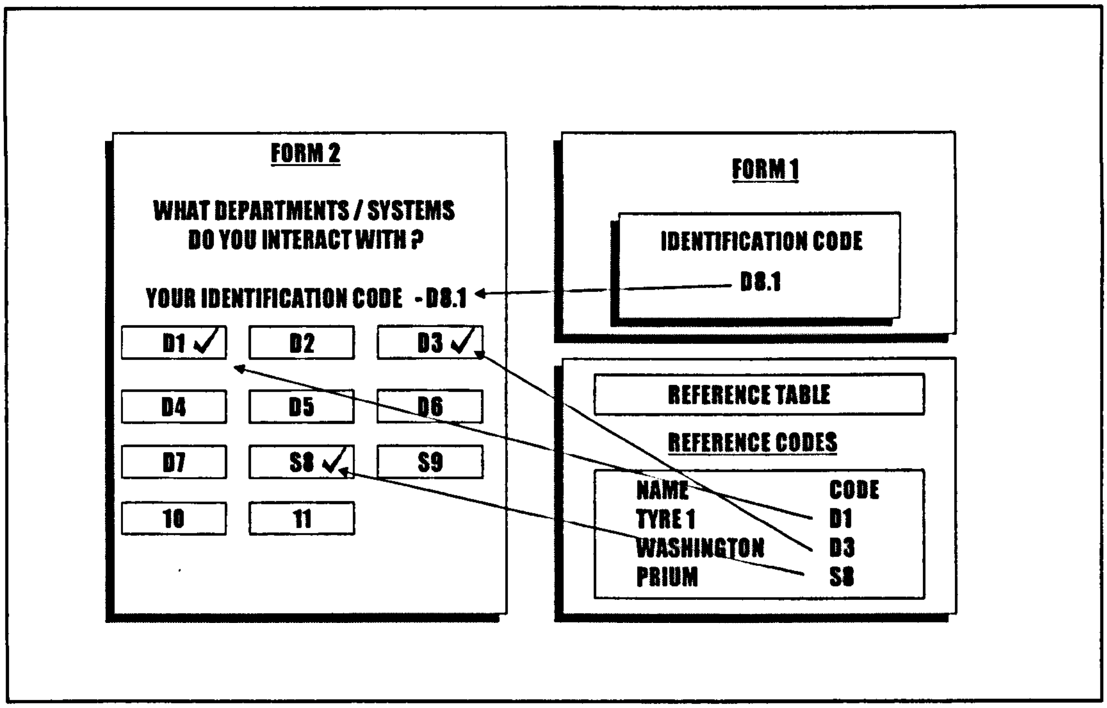


Figure 5.11: Form 3 with reference to forms 1 and form 2

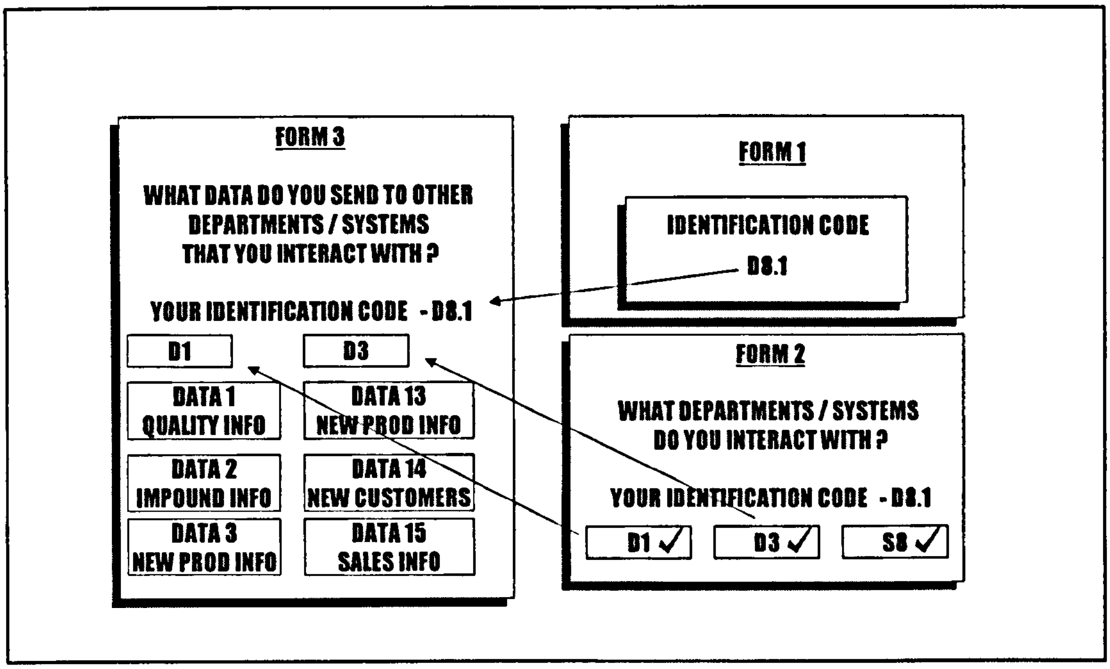


Figure 5.12: Form 4 with reference to form 1 and form 2

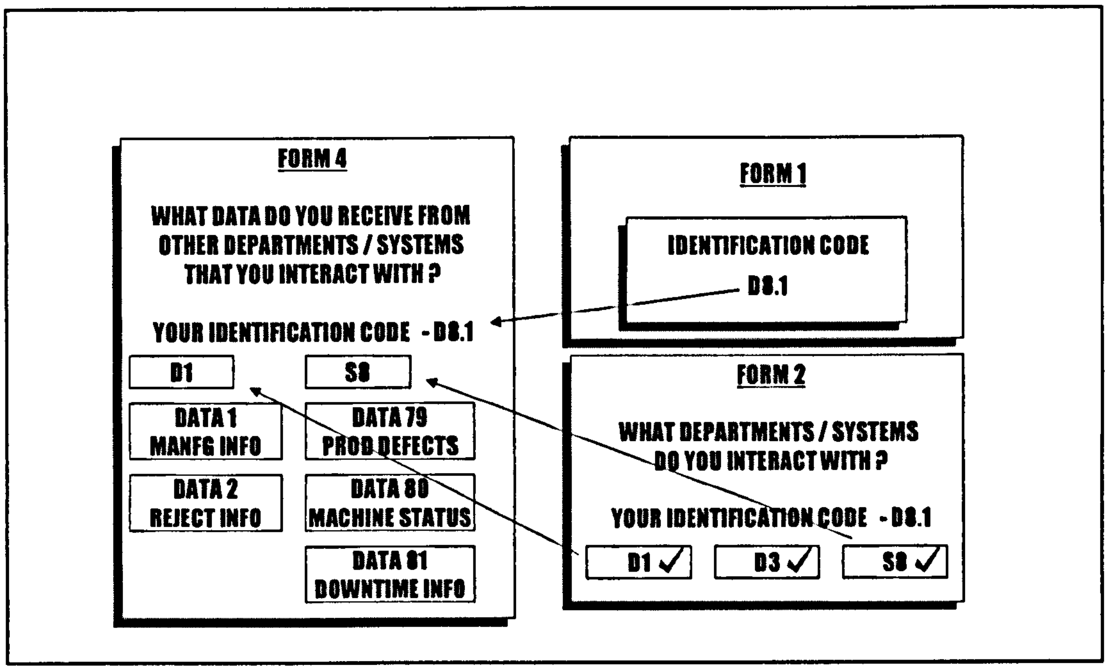


Figure 5.13: Form 5 with reference to form 4

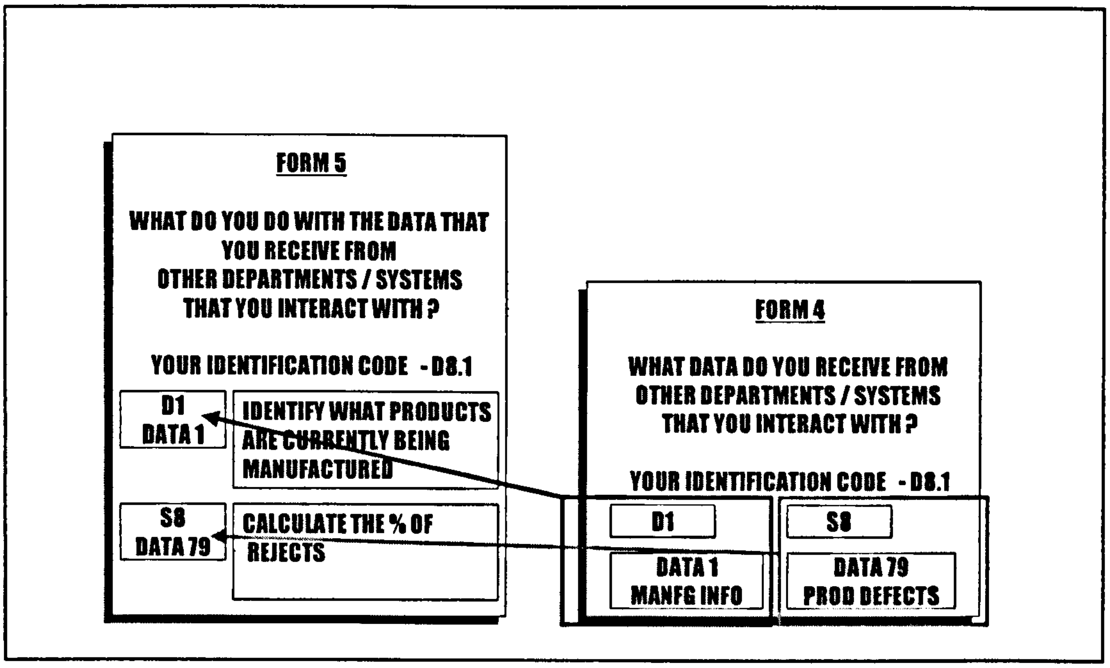
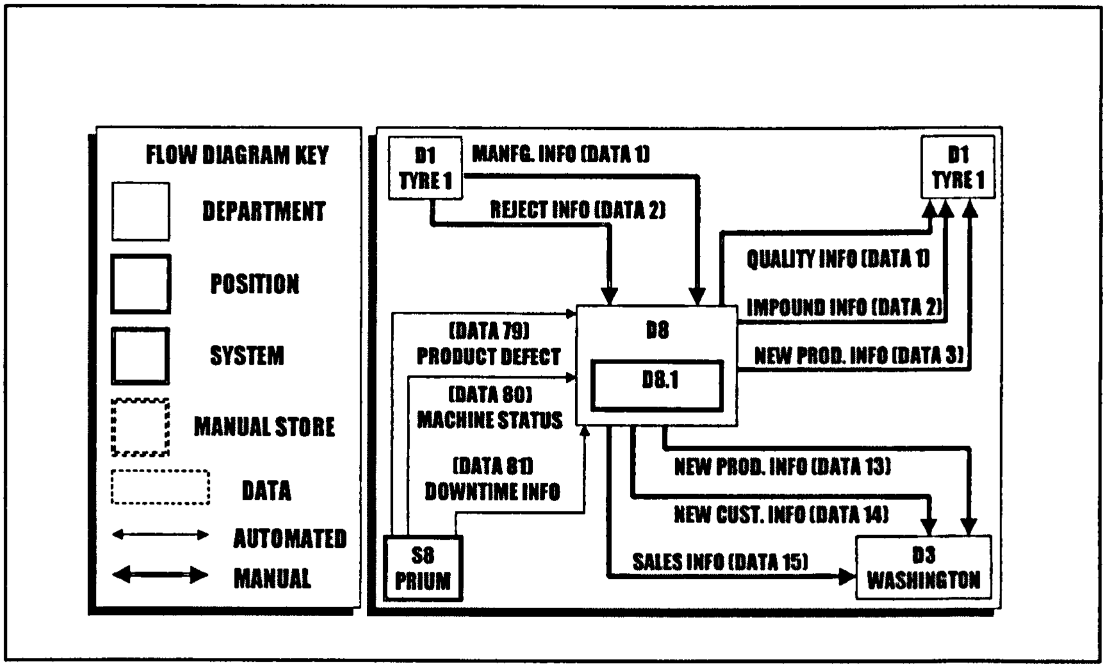


Figure 5.14: Producing the flow diagram based on the information received from the users



5.4.3 The Requirements capturing phases

As – Is (present system)

Understanding the current system

<u>Phase 1</u>

Expectations

Developer – user group

Go through the forms (URCAM forms, appendix D) with the user group and let them know what is expected and how they work

<u>Phase 2</u>

Questions and answer session

Developer – user group

Go through a basic question and answer session on the process. Once the users are happy with how it works go to stage 3

<u>Phase 3</u>

Initial completion of forms

User group

Ensure stationery is *available e.g. pencils and rubbers.* Make sure that these are always used rather than pens as mistakes are likely to be made.

Let the user group fill in the forms together in the same room without the developer. Allow them to write down any questions on a separate sheet of paper for the developer (this should encourages independence and trust). Time the process so that the adequate time can be calculated for future reference.

Phase 4

Review forms

Developer – user group

Developer re-enters the room after a given time. Go through the forms together and check to see if there needs to be any amendments or corrections. Once the time specified is over go to stage 5. (Arrange another meeting with the whole team before anyone leaves).

Phase 5

Users take charge of forms

Users

Users will take the forms with them and look over the forms as much as they wish. They will then be ready for the next meeting with any additions or changes.

Phase 6

Add changes to forms and store for later usage

Developer – user group

Meet again in the same place if possible, as soon as possible to go over the forms with the users adding the changes that have been suggested, once they are feasible.

At this point both the developers and users must sign off the specifications.

Should – Be (proposed system)

Creating the new system

Phase 7

Identify what should be happening

Developer – user group

Developer and user group will now go to the Should – Be section where they look at how the system should operate. This will be carried out in the room in the same way that phases 1-6 were carried out but instead of looking at what is happening now, they should be looking at what should be happening

Phase 8

Users take charge of forms

User group

Users take the forms and think about and discuss the changes that should be made, or identify what should be happening.

Phase 9

Discussion and initial sign off

Developer – user group

Meet again and discuss the changes to the forms and then sign off once everyone is happy.

Phase 10

Developer analyses information

Developer

Developer starts to analyse the information that has been received, looking for duplications and inconsistencies. Then begins to develop prototypes in a more technical format of the information received from the forms.

The developer will be knowledgeable in drawing flow diagrams and should represent the system in a way that the system developers can produce some form of prototype of the information, e.g. on screen dummy runs or some form of manual representation of the screens.

Phase 11

Developer builds prototype and reviews it with users

Developer – user group

Developer brings the prototype and checks this with the users. All information shown should be explained in detail.

The prototypes will be discussed and necessary changes can be suggested

Phase 12

Developer re-develops prototype

Developer

Redevelop the prototype with the changes necessary.

Phase 13

Continue feedback process until agreement is reached

Developer – user group

Go back to users with the prototype after given periods until the time has run out or the users are happy with what has been produced.

Phase 14

Obtain final user approval and sign off specification

Developer – user group

The end result must lead to another sign off by the users and developers ensuring that everyone is happy with the following:

- *Functionality*
- *Ability to find the functionality*
- *User friendliness*
- *Help screen (if possible)*
- *Look of the system*

5.5 The relationship between the literature and the survey of users and developers to URCAM

The specification of URCAM has been developed over a period of time using techniques such as interviews with developers and users, real project experience and an investigation into the existing methods which have been documented by previous writers.

These methods have been chosen for many reasons, these include:

- Their use within industry
- Their use of diagrams to represent data received
- The formal structure that they use

This section of the study will look at how the structured methods have played a part in this development. This will allow the reader to relate some of the thoughts back to the examples where the other methods have been used, thereby adding to the validity of the proposed method.

The survey of users and developers attempted to identify whether there was a need for a new method of capturing user requirements and if there was a need, then the study would attempt to find out what was required from a new method. The results suggested that there was a need for a new method of capturing user requirements and the various factors which have been explained were taken into consideration when the method was developed.

Structured Brainstorming, which was developed by Byrne and Barlow (1993), has identified some points that relate to the development of URCAM. One of the major priorities of the method is for there to be some form of structure to the process. They also ensure that while there is structure there must also be an attitude towards thinking into the future. This relates to the 'Should – Be' section of the process, where the current system is documented and the proposed system must then be identified taking into consideration what has been learnt from the results of the first set of stages.

One of the major aspects of structured brainstorming that has been stated by Byrne and Barlow (1993) is that the method should attempt to 'overcome resistance to change or enhance creativity'. The method's structure attempts to encourage the user to be involved more. This should lead to less resistance and eventually lead to more effective involvement.

Object Oriented Analysis (OOA) has been very influential in the development of the proposed method. The technique is used to model the system using diagrams. This modelling technique breaks down the systems requirements into sections that can be related to each other and looked at in more detail. If we look at the proposed method, we see that a different diagrammatic feature represents each area of the business. For example a box with a thin line represents a department and a box with a double line represents a system. This helps to develop the diagrams for the documentation. Wainwright et al (1996) states that the main concerns of OOA are the identification of significant entities, and the understanding of the interaction among them.

One of the other attributes of OOA is that it divides information into more manageable areas. This is shown in the third activity in the OOA sequence which is identifying subjects, here relevant areas are grouped in the same way that the different levels are grouped.

The Yourdon Systems Method (YSM) has one vital element which links to the proposed method. This is shown in the YSM lifecycle, the element is the System Essential Model. This is used to capture requirements that are extracted from the user. Within the proposed method the aim is to keep the process as basic as possible ensuring that the user can understand the process and have a basic memory of what they do from day to day. Therefore, the interaction between the developer and the user is non-technical.

Within the IDEF0 method there is great importance placed on the people and their involvement in the system. This is heavily stressed within URCAM. The users role aims to be increased by allowing them to answer questions relating to their job rather than using higher levels (only) to carry out the process. However, the IDEF0 method uses a top-down approach, this is seen in one particular area of the method to make the process quicker. An example of this is asking the senior management what departments or systems etc are in operation within the company, then it is left to the user to identify which of the departments they interact with or if they have any to add they can also do that.

Breaking down the 'As-Is' and the 'Should-Be' is very important in the proposed method. The IDEF0 method has been helpful in defining how they should be carried out. Phases are used in URCAM to help form a life cycle for the whole process. URCAM improves on these and greater detail is given that can assist the developers understanding of the company's operations and also assists in carrying out the group sessions.

Doumeingts et al (1993) created GRAI (Graphical Results and Activities Interrelated). This method decomposes the system into different elements that have also been used in URCAM. There are three different elements. These include i) physical, ii) decision making and iii) information. Within URCAM these elements are used. For example, the physical element is represented by the diagrammatic representation, the decision element is shown where questions are asked which help to decide the 'should be' system (proposed system), and the information would be represented by the questions asked to the users which extracts information about the processes that they are involved in.

The GRAI method also has a set of stages that are used to carry out the whole development process. These stages are used in the proposed method. The first set of stages show the most relevance to capturing user requirements as they encourage the developer to identify the management team involved and also to build a management group. Ridgway (1995) states that every diagram or modification is verified and evaluated by the synthesis group (management group). This is very similar to URCAM as users and developers should reach on agreement.

The RAD (Rapid Application Development) method has also been very influential in the development of the proposed method. It has some very useful factors that relate to the interaction of users in the development process. These are as follows:

- Use of workshops – This brings the developer and the users in the same environment allowing a group situation to be formed which aims to help the user to be more confident with their ideas. It also helps users to have some back up when they forget various parts of their job.

- Use of technology – In the case of a RAD workshop; the JAD (Joint Application Design) is used to produce diagrams to represent the system as it is being developed. If the right CASE (Computer Aided Software Engineering) tool is used then this process can be very effective. However, this is very technology driven and can be ineffective if users are not very confident with the idea of using Information Technology.

- Obtain quick results – The purpose of bringing the users into the same room means that the processes can be walked through with all members present rather than individual meetings with each user.

- Higher level of user involvement – This is one of the most important aspects of the proposed method, and the RAD technique has shown this.

The first principle in the DSDM method is that 'Active user involvement is Imperative'. This is a major factor in the proposed method. The teamwork aspect is also one that stands out in the DSDM method; users are seen as teams that in normal cases would mean that they have some common goal. In the proposed method the users are put together in teams of departments or positions and can work together to document the processes that they are involved in.

5.6 Conclusion

To conclude this section, it is important that we realise that the existing methods that have been investigated have been very useful in the development of the proposed method. Various stages and phases have been modified to suit the needs of the study. However, the main factors that relate to most of the methods is that they identify areas where the user can fit into the process, but do not give enough detail as to how they should be involved. However, it must also be pointed out that the methods would benefit from adopting some techniques which will help the process of capturing user requirements to be fulfilled and form some more assistance in obtaining requirements from users. If this is not in the developers scope then it would be necessary to make a point about this fact and direct the user of the method, (for example, SSADM) towards the URCAM method to assist them in interacting with the users.

The most influential method has been the RAD method as it encourages great enthusiasm towards involving the user. It is a good technique for bringing users together and obtaining results. However the questioning process and the method of obtaining these requirements seems to be a secret and therefore is left to professionals to carry out the workshop routine. This process also requires a great deal of administration, which means that the users need to be able to get away from their usual job for a given period of time. There would also be a need to use a range of users at the same time. The decision would be in the hands of the management to see the benefits of the process and decide whether they will be willing to let users take part in the process.

Figure 5.1 shows some factors that relate to user involvement from a developer point of view. It is clear that the developers do feel that user involvement should be increased. They have given some useful tips towards how they think that the method should be developed (see chapter 4). The major aspects of the proposed method that will be directly taken from this area of the study are the initial thoughts (see figure 5.4). These are very important, as they will help the developer to think more before they go into the process. This can also form a checklist which developers can use to assist their preparation stages.

The problems mentioned do show that despite all the good ideas included in the existing methods, there are still major problems with what is happening now. If they are not dealt with, developers will continue to have the same problems with inadequate systems in the future. One of the main considerations should be a drive toward maintenance and longer-term thinking.

The main contribution relates to the soliciting of information from users, which would be used to represent the data being processed within the system. The use of various questions which not only ask what you do but also why you do it, would be beneficial in analysing the information, so that more useful systems are developed. This assists in identify the 'should-be' or 'proposed system' specification. It also means that once the reason is valid as to why a user needs or receives a particular piece of data, then it is likely to be the same in the proposed system.

CHAPTER 6

THE EVALUATION OF URCAM (USER-ORIENTED REQUIREMENTS CAPTURING METHOD)

6.0 Introduction

During Chapter 5 (The development of URCAM), the new method of involving users was described. According to the research methodology, an evaluation process is required to validate whether the method addresses the problems that are being faced in industry.

The purpose of an evaluation as stated by Heyink and Tymstra (1993) should not be to prove a hypothesis, rather to demonstrate that it is plausible. In this case an attempt is made to evaluate what effects the new method would have on the process of information systems development. Weingand (1993) who has a similar suggestion states that instead of beginning with a hypothesis that must be proven either true or false, casting a broad net for data allows the theory to emerge over the process of data-collection grounding the development theory in data itself. Therefore, the URCAM evaluation will enable results to be gained to form a general conclusion of the usefulness of the method more specifically within the process of capturing user requirements.

In this chapter, a number of evaluation methods will be used to draw conclusions about URCAM. This comes from the technique called triangulation, which has been discussed in Chapter 3. The first technique is clustering. This will help to bring like data with like data from different participants in the evaluation process. The evaluation methods that will be used include: NIMSAD, Information Economics and expert opinion. These different methods of evaluation should assist in the process of identifying the various strong points and also the weak points that the proposed method has.

The validation process was assisted by various users and developers from different companies. The companies who took part in this stage of the study were Cranfield University, BICC General Cables, Ford, Sitel Corporation, Dunlop and Pedigree Masterfoods. They assisted by either their developers evaluating the method and/or their users using the method. Questionnaires were filled in once either process was completed so that conclusions could be drawn from the experiment.

6.1 Evaluation design

Czaja and Blair (1995) have identified five general stages in the development and completion of a survey. These are as follows:

1. Survey design and preliminary planning
2. Pretesting
3. Final survey design and planning
4. Data collection
5. Data coding, data-file construction, analysis, and final report

Morris et al (1987) has developed some basic steps in developing a performance test for program evaluation. They state the following:

(1) Determine the outcomes to be measured
(2) Develop a blueprint for the test
(3) Write the test items
(4) Review and edit the items
(5) Field test the items
(6) Obtain reliability and validity data

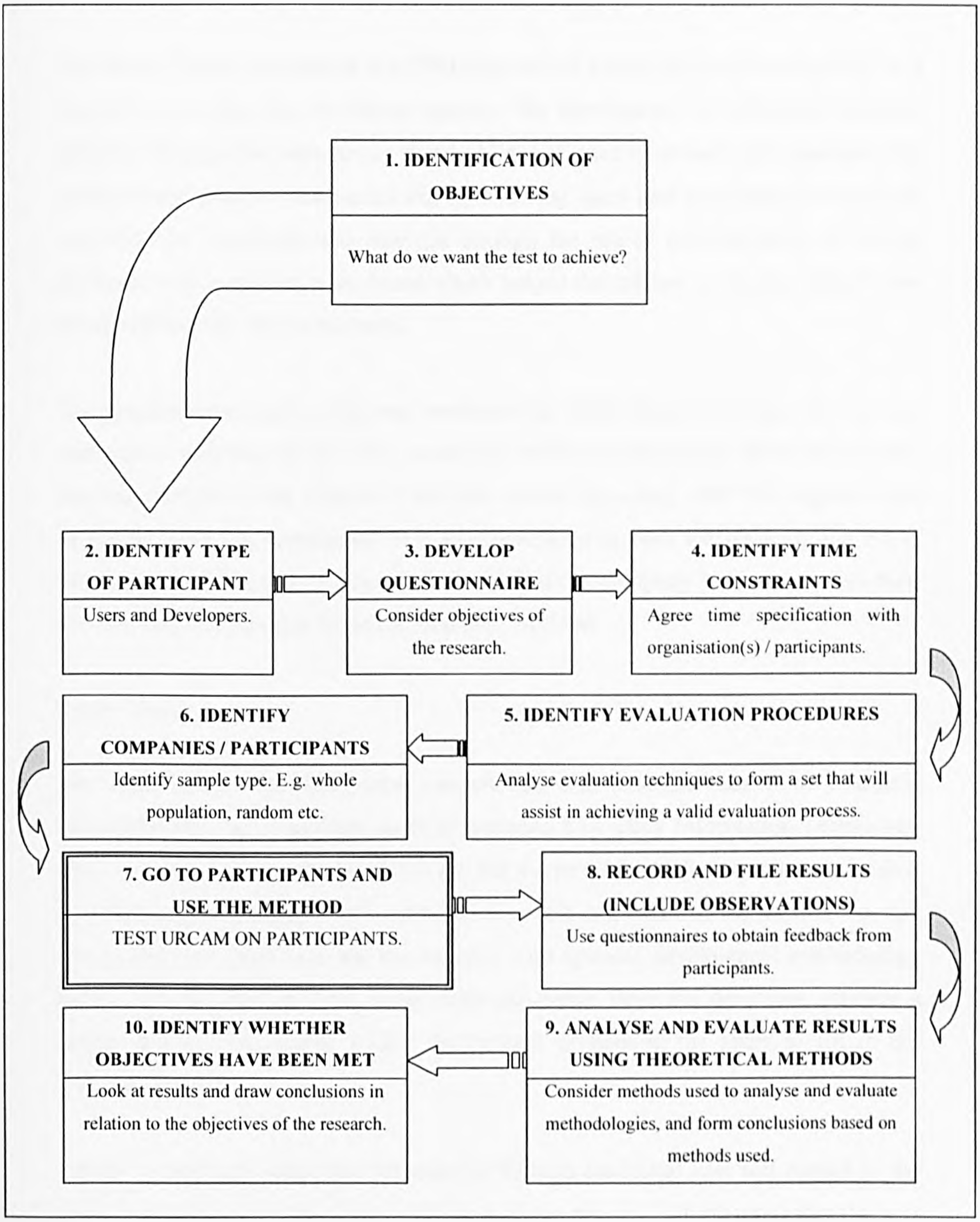
The factors that have been developed by Morris et al (1987) and Czaja and Blair (1995) help the evaluation to identify stages (shown in figure 6.1) that the study should consider before, during and after the test.

Figure 6.1 demonstrates the evaluation design that the study will adopt. It is based on the work of Morris et al (1987), Czaja and Blair (1995). This diagram represents a structure to identify objectives, samples and validity for the evaluation process. The general ideas are based on chapter 3, which specify the techniques used for data collection, analysis and evaluation.

Once the objectives of this evaluation process were clear the questionnaires shown in appendices E and F was created, which took into consideration the factors that were discussed from the investigation into methods of evaluating the research.

Each interview consisted of a structured walkthrough of the method and, in the users cases, they were asked to fill in the URCAM forms and then use the questionnaire to give their opinions about the process and the URCAM development stages. The developers were asked to put themselves in the users shoes and predict their reaction to how they would use the method and what their reactions would be as well as identify what they think of URCAM.

Figure 6.1: Evaluation design



6.2 Initial evaluation

Gundagon (1995) and Dean et al (1994) suggest that validation can be carried out, to a large extent, during the modelling session. The development of URCAM has been achieved through the experiences that had been learned from each test situation. The initial testing process was carried out by allowing users and developers to learn and use URCAM. Feedback was received through the use of questionnaires. From the findings, various factors were found which helped the method to evolve while it was being validated by the participants.

The systems investigation that was conducted in BICC General Cables (site survey) was used to generate results from a site test, which is explained in more detail in the site test section of this chapter. This was carried out using URCAM together with verbal interviewing techniques. This was beneficial to both the research and BICC General Cables because it generated results that the company used to improve their systems and findings that helped to improve URCAM.

6.2.1 Users

The initial tests with users were carried out with a single user from Cranfield University who had a medium level of experience of using Information Technology (IT). The user was expected to fill in the full set of URCAM forms with very limited help from the developer. The conclusion from this test was that the method was too complicated for users who are not familiar with systems development methodology usage. This resulted in there being more assistance from the developer offering a higher degree of guidance within the process of helping the users to fill in the URCAM forms.

Another important factor that was learned through the initial user test related to the size of the boxes (they were too small to hold the required information) that the user used to fill in the relevant information. This may seem like a minor issue, but it is very important for the clarity of the information provided. One important objective of

URCAM is to improve the quality of the information obtained from the users. Therefore, the users need to be able to express their information in the correct format so that it can be understood by the developers. However, there is still a need to keep these areas to a limited size as too much information may detract the developer from the relevant points and may also take too much time. An extra box titled name is also added. This is so that users do not have to remember all of the codes that are used in URCAM.

6.2.2 Site Test

The purpose of this exercise was to test the use of URCAM (User-Oriented Requirements Capturing Method) by collecting information from users of information systems and walking through the method with developers. The ideal test would be to have a set of users allocated to the systems development exercise. This would mean that the different stages of URCAM could be used to extract the information from the users and then analyse the information to develop a new system, which would allow each stage of URCAM to be tested. However, this was not possible due to management not giving the researcher access to the users, due to time constraints as the company was going through a merger.

However, due to problems and complaints being logged by the users of the existing systems within the Wrexham 1 factory, the BICC General Cables manufacturing systems department decided to carry out an investigation (a site survey) to identify exactly what the problems were and how the problems could be solved. Depending on the results of the site survey, changes within the functionality offered by the existing information systems could be made. This would result in the factory information systems being able to produce more useful information for its users.

The management within BICC General Cables authorised the researcher to participate in the site survey as a member of the development team (who are members of the manufacturing systems department) carrying out the investigation. This presented an opportunity for URCAM to be tested, as access to members of the factory would be required to carry out the investigation. However, the time specified to interview the users was around 30 minutes and this did not give enough time for the whole method to be tested. Therefore, a selection of forms were chosen that could be used within the time limit given.

The interviewing process did not have the conditions where the proposed method could be carried out in a very formal format. The testing was carried out fitting around the day-to-day running of the factory, causing the least amount of disruption to the users' jobs as possible.

URCAM would be used together with interviewing to obtain information from users during the site survey. The users would be interviewed in groups or individually depending on the number of users that are involved in the site survey. This should test how users react in groups or individually.

The test will also see if URCAM will produce relevant information for the manufacturing systems department and observe how users respond to the forms used.

Site survey specification

The site survey used a collection of users of the existing systems. These systems included the CFMS, AMPLE and ALMS which are all in use in the BICC General Cables Wrexham 1 factory. The users consisted of process managers, team leaders and operators who all interact with the systems. The positions that the users held were as follows:

Process Manager Wire Drawing / Stranding
Process Manager Monosil / PVC
Monosil / PVC Extrusion Line Operator
Project Engineer
Process Manager Wire Drawing / Stranding
Wire Drawing Operator

Monosil / PVC Extrusion Line Team leader

Oversheathing Team leader

Laying Up Team Leader

Laying Up Operator

Armouring Team Leader

The 11 users of the factory's information systems were interviewed. Each interviewee was asked the same questions. Every user involved was given the opportunity to voice their opinions about the functions that they carried out each day, how they used the factory information systems, what they thought of the factory information systems and how they thought that improvements could be made.

The questions were developed from looking at the problems that had been logged by users before the site survey. The main questions asked covered the following areas:

- Employee position
- Activities / Responsibilities that they carry out
- Information sent and received
- Reports, Forms sent, received and / or filled in.
- OEE calculations and its importance
- Machine Speed and Status codes
- Drum Resets
- Functionality and usage of AMPLE, ALMS and CFMS
- Interprocess testing
- Operator Logging
- Calibration dates
- Die / valve set maintenance and rejection management (scrap management)

Findings of the site survey

The manufacturing systems department have used the information collected within the site survey to initiate changes in the systems that relate to the following points:

- The shop floor workers log the status of machines at specified times, for example if a machine breaks down. However, the records of the faults do not seem to find their way to management, who need to know this information.

- The users expressed that they would like to have more information regarding the usefulness of the reports and the information that is generated.

- There were a few points that seemed to cause some concern among the users. These included the fact that there was a lot of manual transactions such as forms being filled in and then sent somewhere, but there was a question about what the figures produced are being used for. The main complaint being a lack of feedback.

- The users feel that automating the process of inputting the machine status codes would be useful.

- There is a need for some further information that is requested via the use of the CFMS.

- There is a need for users to be able to input reasons why machines have stopped for any particular reason. For example if the machines have stopped, currently the CFMS simply shows a status showing that machines have stopped. This is not enough information for further analysis, for example if a regular occurrence happens then the process manager needs to have detailed information about where to direct resources to stop certain things from happening.

- Alarm bells should be triggered if machines are running at the speed not specified within the product recipes. This would mean that the process manager could deal with the problem immediately rather than a product taking longer to be produced.

Lessons Learned from using URCAM during the site test

Users were interviewed in groups of two and individually. The group situation did produce more enthusiasm and users seemed to be able to express themselves more and help each other to recall various elements of the systems. Individually, they still showed some enthusiasm but were not as positive about the information that they gave.

The interviewing sessions consisted of users being asked questions that they answered verbally. The evaluation process attempted to see how users felt about writing down information on forms. Therefore, they read questions and then filled in boxes with written information. The users were also asked to read questions and tick boxes. These methods of capturing user requirements were looked at in more detail as the test continued.

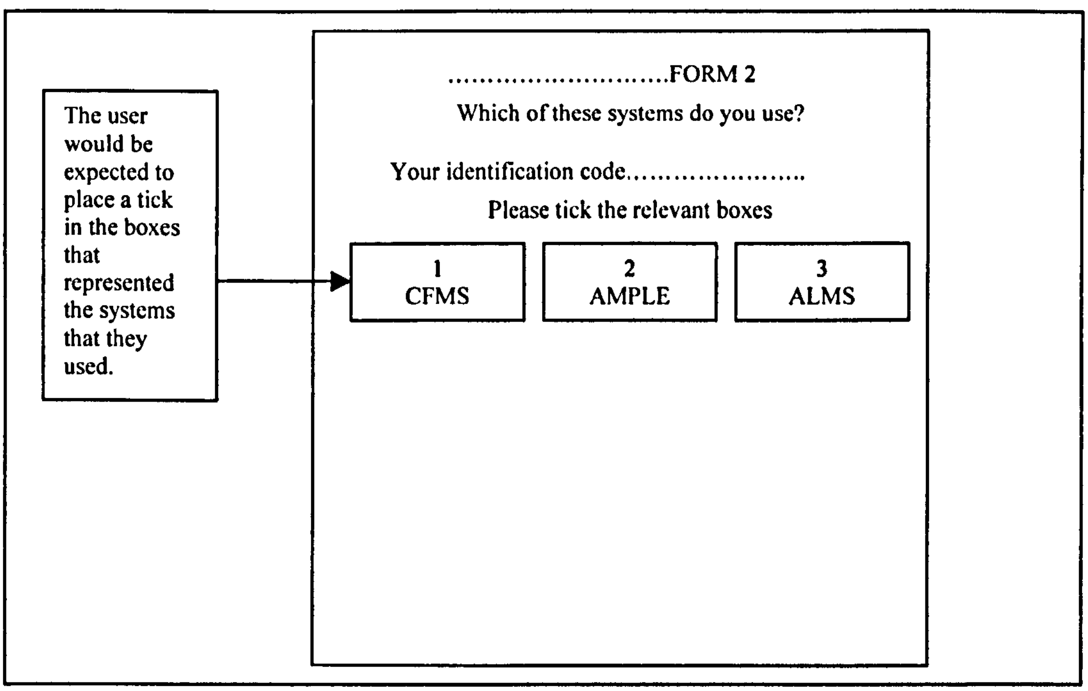
<u>Verbal Communication</u>

The interviewer asks questions and the interviewee will answer the question verbally while the interviewer writes notes according to what the interviewee says. This method can be effective and allows the user to express their thoughts freely. However, this method is dependent on the interviewer being able to write fast enough and to clearly understand what has been said by the user. Tape recording can be used. However, there may be objections to this method. This correlates with the research that was carried out by Ruebenstein et al (1991) who identified various factors such as abbreviation, ambiguity etc. that are results of verbal communication (this was explained in more detail in chapter 2).

Ticking Boxes

Form 2 was used which is shown in figure 6.2. This form requires the user to tick a box which contains the options that they felt answered the question that they were asked. In this case the question was, "Which of these systems do you use?" This was used to allow the development teams to obtain an understanding of what systems the users were aware of, and what specifications they were interested in. In this method the information was already provided and the user had a simple task of just ticking the box that was relevant to them. After analysing this method, the conclusion obtained was that the method is good for identifying very brief or direct questions, but it encourages the user to only consider what options they can see rather than thinking further, which may produce more effective information.

Figure 6.2: Diagram showing how users were required to tick boxes



Filling in boxes with descriptive information

Lastly the users were asked to fill in boxes stating what information they received or sent, to or from different systems. In this case the user was expected to remember the systems name, write it down and then write down what information they sent or received from or to that particular system. What was noticed is that the user needed to spend more time thinking about what they needed to write down and seemed more concerned about the information that they gave.

At the end of each of the sessions the users answered questions about the use of the forms and the development process. The results from the questions, which are shown in table 6.1, suggest that users were very enthusiastic about being involved in the process. All of the users who were interviewed had taken part in a process where they were asked to recall information regarding their day-to-day job functions. This meant that they had some knowledge of how things were done in the past. It was clear from speaking to the users that in the past they had not been involved as much and systems were given to them and they were expected to use them.

The use of the forms did not seem to cause much difficulty to the users. However it was necessary to explain exactly how to use the forms using the help sheets that are provided in URCAM. All participants understood the forms once they were explained.

The results shown in table 6.1 (based on 11 users) provide some reasons why the proposed method could be developed further. They show that the users who were involved understood the forms and felt that taking part made them feel more involved. One of the users stated that the forms gave the process a more professional approach. This is in line with the aim of the method, which is to formalise the process.

Table 6.1: Table showing answers from the site test

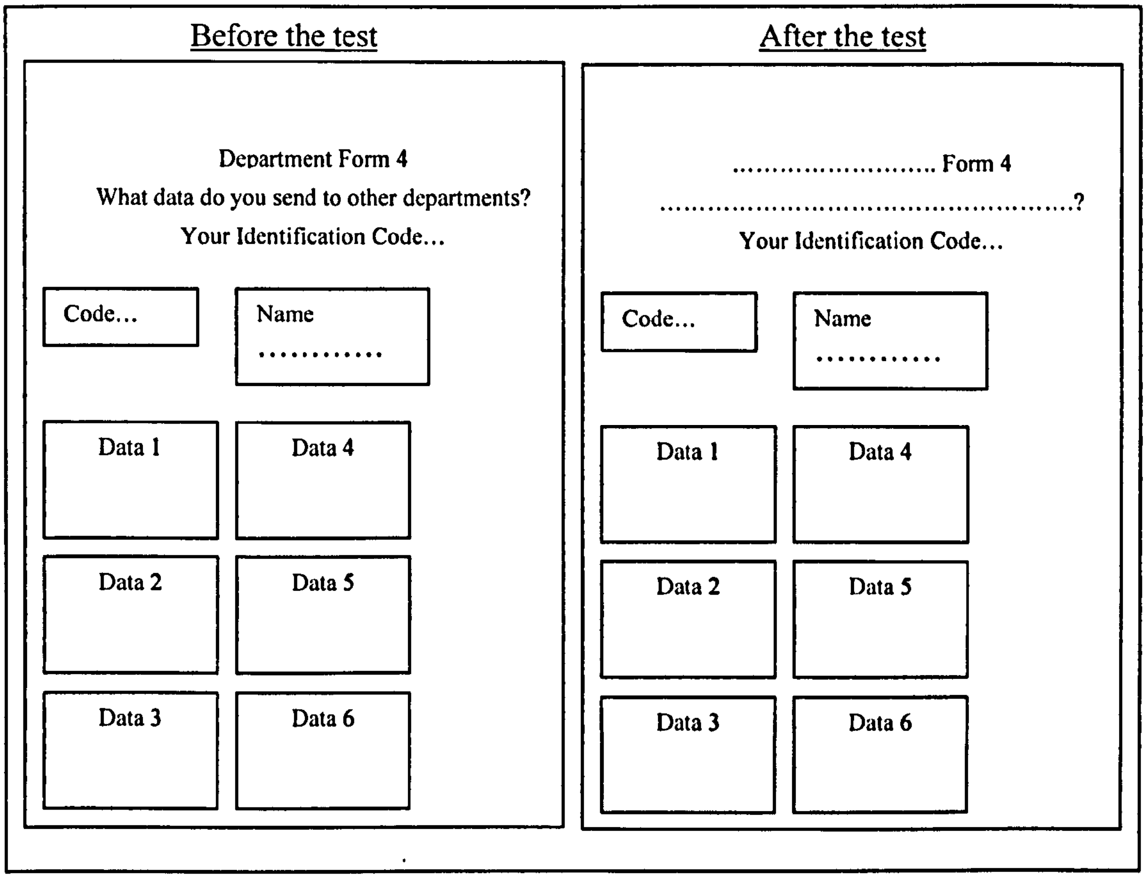
Questions		% who answered yes.	% who answered no.
1. Have you been involved in the process of developing information systems before?		100	0
2. Did you understand the URCAM forms?		100	0
3. Did using the URCAM forms make you feel more involved?		100	0
4. What would be a good incentive for you to get involved in the URCAM process?			
A.	Knowing that you were involved?	55	45
B	Systems meeting requirements?	45	55
C	Some financial reward?	18	82
D	Some social activity?	18	82

URCAM development forms used as templates

Different projects require different information. For example, some projects want to find out what reports a system produces while others may want to find out how often a specific piece of information is received from a system. This means that the forms should not be so rigid that they can only be used to ask one set of questions. This is shown in figure 6.3. Therefore, the development team can decide what questions they should ask on the dotted line and the users answers can be placed within the same boxes provided.

From using and analysing the three methods (verbal communication, ticking boxes and filling in boxes), it was concluded that a combination of the three methods could be used. This will depend on the users level of understanding of URCAM. However, the URCAM forms should be used to keep a structure throughout the process of capturing user requirements. This conclusion is drawn as a result of all methods offering positive points. Therefore, none of them should be omitted. The combination used would be left to the discretion of the developer who will assess the abilities of the users and also have an idea of the time required and the time available to carry out the whole process.

Figure 6.3: The forms used as templates



6.2.3 Expert opinion

This area of the testing period was conducted with experts within the field of Information Systems Development. The reason why they were chosen was that they have worked in the field of Information Systems Development for many years and have had experience of dealing with users and collecting user requirements. Each expert acknowledged that URCAM was very useful. However, they had some valid questions which were taken into consideration when redeveloping URCAM. These are discussed in this section of the thesis. The experts addressed the issues that they felt were important to consider in capturing user requirements.

The questions from experts are as follows:

1. How will the analysis take place after the information is collected?

(R. Rahmani-Torkman, BICC General Cables)

The forms can be used to ask users a variety of questions. For example, "what do you do with the data that you receive?" For each item of data that they receive, they must be able to let the developer know what happens to that piece of data. If the form shows that there is no constructive use of the data then there will be cause for some concern. The user and the developer will be able to decide whether they need that data or not. There is likely to be some form of report that is sent somewhere, but nothing happens to the report. For example, "it is not used for any further analysis or to make any decisions within the organisation". Alternatively it may be stored in a file and then disregarded after a certain amount of time. The developer needs to know this information so that the system can stop producing that report.

2. Would the method be able to generate any diagrams (for example flow diagrams)?

(D. Wright, BICC General Cables)

A diagrammatic format has been developed for the information that the forms generate. These diagrams specify the difference between departments, positions, systems and manual storage areas. The diagrams also show the flow of information between different systems. These diagrams will be able to detect duplication in information flows and also show who within a department receives and sends various items of data. The diagrams are produced at a simplified level. This would mean that they are more likely to be understood by all levels of users.

3. Why do the forms for data received have the same number as for data sent?

(N. Matupi, BICC General Cables)

The developer has the ability to decide how many of each form is necessary. For example, if the users usually send data rather than receive data then the developer should have more forms that ask the question what data do you send?

4. How will the method deal will large amounts of information?

(R. Rahmani-Torkman)

Each users answers need to be kept individually. This will help the process to be able to hold large amounts of information. Also once the forms are used correctly the information provided should be clear and easy to understand. For example if a system is used, its name should be clearly stated in one box and the information received will be placed in the boxes that correspond with that system.

5. Use of some form of software to represent the data being collected?

(R. Rahmani-Torkman)

The use of a software package would be beneficial to the forms. However, this would be for the developer rather than the user. Users at shop floor level for example may find it hard to type information and may take longer than they would to write down the information required. Forms can be photocopied and used anywhere. If a software device was developed this would mean that equipment would be necessary and this may effect the speed of the interviewing process.

6.3 Main evaluation (re-test)

Once the initial findings were used to redevelop URCAM the next set of evaluations were conducted in a collection of companies including BICC General Cables, Sitel Corporation, Pedigree Masterfoods, Ford and Dunlop. Each company offered the research some form of evaluation of URCAM. The questionnaire was used to form an evaluation of URCAM. The discussions within each evaluation method in this section of the thesis are based on the results of the questionnaire, which a shown in table 6.2.

6.3.1 NIMSAD (Normative Information Model-Based Systems Analysis and Design)

The NIMSAD (see chapter 3, section 3.3.1) method encourages the evaluation process to consider a set of elements to form conclusions about a proposed method. These elements are as follows:

1. The problem situation. Within this study the problem is the lack of user involvement. This is demonstrated by question 14 in the questionnaire (shown in table 6.2), where 43% answered 4 and 43% answered 5 on a scale of 1 to 5 where 1 represents less involved and 5 represents more involved. These results suggest that according to the users, URCAM showed a positive response to the users feeling more involved by using the method.

2. Identify the intended problem solver. In this case the intended problem solver is the developer. According to the developers interviewed they felt that the user is likely to feel more involved. However, their ratings were less than the users. This evaluation was arrived at by developers comments about the intended solution, where they identified that the proposed method would be easy to use and offers a good structure.

3. The problem solving process. URCAM was the problem solving process. There were various evaluation questions that were used to test what developers thought

of URCAM. The comments given suggest that URCAM has a very good structure for carrying out the process of capturing user requirements. This would help developers to go through this particular process in stages allowing them to use their experience together with a structure that they can always refer back to.

6.3.2 Information Economics

Parker et al (1989) state that information (see chapter 3, section 3.3.2) is categorised as having measures of worth other than monetary measures. Each section of the questionnaire was used to achieve a specified outcome about the proposed method which had no direct relation to financial rewards. The sections were as follows:

1. Previous experience
2. Usability of the forms
3. Effectiveness of the information gained from using the forms
4. Usage of help sheets
5. Comfort with filling in the forms
6. Effects on involvement

The sections above (1-6) contained questions that were answered using ratings from 1 to 5.

1. Previous experience

The questions on previous experience were used to find out if users had been involved before and what methods where used. The scales were used for identifying the confidence that users felt once they had given the information during their previous experiences. The responses ranged from 2 to 5, where most of the participants said that they were confident that what they said would go into the new system. Question 4, which is shown in table 6.2 suggests that what the users expected and what happened was quite different, most said that only some of what they said went into the new system.

2. Usability of forms

The questions on usability of the forms was answered by participants giving them room to identify the good points and the bad points about the URCAM forms. The good points where very similar and generally pointed towards the forms being simple and a useful form of documentation for capturing user requirements and very useful for tracibility of information. The bad points related to the forms needing to be more flexible and the need for backup procedures to make the information more meaningful.

3. Effectiveness of the information gained from using the forms

The questions on the effectiveness of the information gained from using the forms consisted of three questions. The first was "How does the information that you have given on the forms compare with the information you would have given verbally?" Again the scale was used from 1 to 5 where 1 is worse and 5 is better. The answers that were given ranged from 3 to 5, where the highest percentage showed that the information given using the forms would be better than what is given verbally. However, most of the answers were 4s and 5s. The second question was, "How confident are you that the information that you have given is going to be used because you have written down your requirements?" The scale was used again where 1 represent not at all confident and 5 represents very confident. The answers given ranged from 2 to 4, this showed that the users would feel more confident by filling in the forms that the information would be used. However, as the results suggest, there is still some doubt about the information being used. The last question in this section looked at the time taken to fill in the forms. The highest percentage identified that the forms took enough time. This time lasted around half and hour. However, this time will depend on the size of the system that was being investigated.

4. Usage of the help sheets

The proposed method contained a set of help sheets. These were used to demonstrate the forms to the users. Within the evaluation process participants were asked to evaluate their understanding of the help sheets and the usefulness of them. The answers given suggest that the participants had a reasonable understanding of the forms and also found them very useful.

5. Comfort with filling in the forms

Their was a slight sense that participants thought that filling in information and filling in their name might cause some problems if the method was used. This is a factor that requires further research to identify how users can be identified without them writing their names, as this identification is very important for tracibility of information. Information may need to be clarified at a later stage or further information may be necessary from a particular user. Therefore, some form of record of the user becomes very important.

6. *Effect on user involvement*

One of the most important areas of the study is the effect that the forms will have on user involvement. When the participants were asked what the likelihood that users would use the system as a result of user involvement, most of them gave a 4. This suggests that the forms would make the users feel more involved and more confident that the information that they gave as a result of using the forms would be used in the development of the new system.

Table 6.2: Results from main evaluation survey

Section	Question	% that chose a number on a scale of 1 to 5					
		% who did not answer	1	2	3	4	5
Previous Experience	3. Confidence with what information has been given will be used in the new system		Not at all		Confident		Very confident
		14	0	29	43	0	14
	4. Expectation that what you expected to be used was used		None		Some		All
		14	0	14	43	14	14
Effectiveness of information gained from using the forms	6. Information given on forms compared to verbal communication		Worse		The same		Better
		14	0	0	14	29	43
	7. Confidence that what you wrote on form will be used in the new system		Not at all		Confident		Very confident
		14	0	14	29	43	0
	8. Time required to complete form		Too short		Enough		Too long
		29	0	0	43	29	0
Usage of help sheets	9. Understand the help sheets		Not at all		Reasonably		Totally
		14	0	0	43	43	0
	10. Usefulness of the help sheets		Not at all		Reasonably		Very
		14	0	0	0	29	57
Comfort with filling in forms	11. Fear of writing down information		No fear		Cautious		Very fearful
		14	29	29	29	0	0
	12. Fear of writing your name down		No fear		Cautious		Very fearful
		14	43	29	14	0	0
Effects on involvement	13. Likelihood that system will be used as a result of involvement		Less		No different		More
		14	0	0	14	57	14
	14. How involved does filling in the forms make you feel		Less		No different		More
		14	0	0	0	43	43
	15. Confidence after filling in forms that information given will be used		Less		No different		More
		14	0	0	14	43	29

6.3.3 Expert Opinion

The industrial visits were used to obtain some feedback of how useful URCAM would be when used in a company. The questionnaire was also able to identify some descriptive statements from experts about URCAM. The experts used in this section of the thesis have also had a great deal of experience in the development of information systems and capturing user requirements.

The experts helped to identify some validity of URCAM, as it was not possible for the method to be fully tested, where only the results gained from the use of URCAM were used to develop a system. Therefore, the participants were used to give the research the required feedback.

The comments given by the experts (in italics) are followed by notes from the discussions of the comments that were carried out with the experts.

The feedback that was received from the participants was positive.

These are some of the comments that were made:

1. *Introduction into any new system is so important for a low level user* (Mr. Larson, Pedigree Master Foods).This pointer is emphasised in the development stages of URCAM. The developer will be encouraged to invite as many users as possible to the initial meeting that explains why the system is to be implemented or why the process of change is taking place.

2. *It is useful to know about the system when it comes to implementation and commissioning or training* (Mr. Larson). The initial meeting proposed in URCAM will cover this point. There seemed to be an importance from a user point of view that they need to know what is happening and why it is happening. This seems to be a reasonable request so that users know what they are involved in producing.

3. *Things written are reproduced better over the long term than verbal* (Mr McMullins, Sitel Corporation). There is a positive sign that users writing down what they consider to be true would be beneficial to the development process. The need for long term thinking may encourage better documentation of information so that at later stages detailed information can be recalled rather than having to go over the same process again. Therefore, the developers can concentrate on identifying what they want to happen e.g. the 'should be' section of the development stages.

4. *Forms must be tuned to specific applications* (Mr Stenzel, Pedigree Masterfoods). This was learned in the initial testing process. The forms became templates where adjustments can be made to represent the desired information from the user.

5. *Good idea for self-supporting groups of users* (Mr Stenzel). This means that the groups can work easier on their own with URCAM's structured approach. The help sheets are provided so that users can refer to them in times of difficulty.

6. *Opportunities for people doing the same sort of work to enable group support* (Mr Stenzel). This point should encourage users to work together. However, the individual usage of the forms is still possible.

7. *Method enables more time for thinking* (Mr Wallis, Pedigree Masterfoods). The users recognised in the test cases that they have slight concerns about writing down information. This could be for a number of reasons including the fear of them being asked questions at a later stage about what they have written. This should encourage the user to write down more precise information which should take more time to give. This should be taken into account when the time is specified for how long the process should take.

8. *Able to review documentation* (Mr Wallis). The developer can always say that the user wrote the information down and not them. This also relates to the user being able to say that a developer needs to show them what they think that they said. If it

is not written down then there is no proof that the user said what the developer has written down.

9. *Forms make users think more* (Mr. Mahmood, Dunlop). This relates to the user not wanting to make mistakes. This should encourage them to take more time and think harder about what they are going to write.

10. *Information stated as the user wants it to be stated rather than the interviewer interpreting the information in their own way* (Mr Mahmood). This is a very important point as verbal communication can be misinterpreted which slows down the development process as systems have to be re-written due to inaccurate specifications. The method encourages the user to write down exactly what they want to be included.

11. *Method could be applied to other data collection exercises* (Mr. Mahmood). The method could be used to collect data about anything that needs to be changed, for example, what parts are required to build a particular product. This also relates to the forms being used as templates.

12. *User friendliness of the method is very important* (Mr. Mahmood). It is very important for the method to be able to be understood by different levels of users from shop floor to management levels or less experienced to experienced participants of systems development exercises. The simplicity would encourage less experienced users to get involved in the process.

13. *Method could be improved by thinking of a backup structure or the next stage in the process* (Mr Crowdell, Ford). The back up structure could be an automated system which would be more useful for very large amounts of data, where the information collected at different stages could be inputted in the system, for example, a database so that information can be recalled at any time.

14. *Think about how to represent the information received in a visual format for example using post-it notes on a large notice board to show flows of information through the company* (Mr. Crowdell, Ford). The idea of using post-it notes offers a good factor for further study. This would help to represent the information so that users can see it clearly and changes can be made to the flow of information physically without the use of computer technology to print out diagrams that users may not understand.

15. *Method offers a very good opportunity to trace the information that has been received* (Mr. Crowdell, Ford). The different codes used within the proposed method allow each item of data to be traced back to its origin. This would be very useful for developers who are creating the system and for analysis purposes, where developers can see where each item of data is coming from and where it should go to.

6.4 Conclusions

During the initial evaluation, users filled in the URCAM forms and then answered questions that enabled the research to form a deeper understanding of their feelings towards the method. The expert opinion was useful as it gave an industrial point of view. Each opinion offered the study suggestions for improvements to URCAM.

The interviews conducted during the site survey tested the use of verbal communication, ticking boxes and filling in descriptive information. The different areas within the factory were involved to generate the initial findings and were able to voice their opinions about the functionality of the systems that they were using. The method was able to get the users to think about the work that they currently carried out each day and write it down so that it could be used to improve the existing system. We can see from the findings of the site survey that the mixture of the methods was able to generate the information that was required to improve the existing systems, even though the whole of URCAM was not used.

After the initial evaluation process, it was clear that the method could still be improved with continuous testing and feedback from further fieldwork. However, the usefulness of URCAM is in its ability to improve the relationship between the user and the developer within the systems development process and for them to gain a better understanding of each other.

The re-test situation was used to validate URCAM. This test was carried out in a collection of companies where users and developers were shown the method and in some cases users used the URCAM development forms. The questionnaires were completed so that feedback from the participants could be analysed and conclusions could be drawn.

The results from the questionnaire formed the basis for drawing the conclusions about the evaluation process. NIMSAD was used to identify the initial structure for the evaluation exercise. This suggests that URCAM was able to address the problems that were identified in the literature and those encountered by industry. Therefore, offering a solution that the participants involved in the evaluation process found useful.

It was possible to observe a weakness in the method in influencing the higher management to accept the method. This may not be a result of the validity of the method, but may be a result of management attitudes to change. This may be researched further to identify more accurate conclusions.

The information economics section was used to evaluate the different sections of the questionnaire where clusters were formed to draw conclusions. The overall assessment of the factors suggest that users and developers are both important participants of the development process. The users would be able to use the method effectively and give more accurate information than if they were giving information verbally. The results also suggests that users would feel more involved if they were to use the proposed method.

The expert opinion suggests that the proposed method addresses some of the important factors that effect the process of capturing user requirements. This is as a result of the industrial interaction which was carried out in the study. The factors obtained from the industrial visits were broken down into key factors which are described in chapter 6 and URCAM takes these into consideration.

Finally, according to the evaluation process it is possible to conclude that the method can offer a solution that will improve the process of capturing user requirements. One of the main weaknesses is gaining acceptance at higher levels of management as URCAM would require a change in attitude towards evaluating methods that are different from normal procedures such as verbal interviewing alone. However, the results from the evaluation process suggest that once users and developers are taught how to use the method then it would be effective in increasing user involvement and improving the information that is obtained by end users.

CHAPTER 7

DISCUSSION AND CONCLUSIONS

7.0 Introduction

This chapter discusses and concludes the findings of the research. The aim of the research was to review the problems associated with information systems that relate to user involvement and to produce a method of capturing user requirements during the systems development process. The research aim was addressed by completing a set of objectives.

The Objectives of the research described in this thesis were as follows:

1. Identify and compare existing systems development methodologies and methods designed to address user requirements. Discuss their strengths and weaknesses in relation to the extent and appropriateness of user involvement within their structure.
2. Investigate the validity of the belief that a lack of user involvement is a major problem that affects systems success or failure.
3. Identify the requirements for a new method of capturing user requirements from both parties (users and developers).
4. Develop a method for capturing user requirements. The method should consider the strengths found in existing methods and the requirements stated by users and developers in industry. Then evaluate the method and identify what the results of the research indicate.

7.1 Discussion and Conclusions

Objective 1

Identify and compare existing systems development methodologies and methods designed to address user requirements. Discuss their strengths and weaknesses in relation to the extent and appropriateness of user involvement within their structure.

A collection of methods (e.g. RAD, DSDM, OOA etc) have been identified and compared. Their strengths and weaknesses have been discussed (see chapter 2). The findings suggest that the methods aim to satisfy specific areas, such as design and analysis. However, results from the literature suggest that even though there are hundreds of commercially available methodologies (Longworth, 1985; Bubenko, 1986), methodology usage is still low (Lubars et al, 1993; Southwell, 1993; Hopker, 1994) and there is still difficulty in achieving systems success, particularly in meeting users needs (Whyte and Bytheway, 1995). The existing methods were reviewed to obtain a general understanding of them and to identify whether they considered the users and if so how. The results (see chapter 2) suggested there was room for improvement in the way that the user was considered. However, the ideal situation was to test each method in a project situation and obtain results from the test to determine how involved the users felt. This was not a feasible option. Therefore, the methods were evaluated based on what other researchers suggested and also what the expectation of users and developers were from the survey of users and developers (see chapter 4). Some methods specify where the user should interact or what an analyst can do with the results. This suggested that different methods could be used to form an amalgamation of ideas to form a new method that incorporates a collection of the benefits from the existing methods (see chapter 5).

When reviewing the literature, the most popular methods (e.g. RAD, DSDM, OOA etc.) were highlighted and gaps in their practical application were found (these are shown in table 2.4 and 2.5). These gaps would represent the area where a new method would fit into the existing body of knowledge. The main points were that the methods were difficult for the user to understand, they had no real structure for involving the users, no structure for capturing the requirements, the user is not seen as a major factor and presented a lack of feedback to users throughout the process. However, there were strengths offered by these methods, which are shown in figure 5.1. The strengths such as using a workshop where groups of users and developer are present to carry out the process using a structured approach, using a social perspective (e.g. understanding user needs), continuous feedback and prototype development were considered to develop a new approach. This led to a further investigation of methods that claimed to involve users or paid more attention to user requirements (see table 2.6). These methods (e.g. Goal modelling and Scenario authoring, use-case, Requirements Apprentice etc.) focused more on the requirements process, where it was clear that they aimed to deal with what to do with the requirements and how they should be developed and represented. There were also gaps in these in that they lacked simplicity and a capturing process (e.g. the user-developer interaction and structure). However, once again, they offered positive points that would be useful in developing a new approach. These points included using Data Flow Diagrams, analysing requirements and formalising the documentation process.

Objective 2

Investigate the validity of the belief that user involvement is a major problem that affects systems success or failure.

User involvement has been shown to be a major problem that affects systems success or failure (see chapter 2). Zwegers et al (1997) state that a major part of the IS design process is the specification of IS requirements, and Ewusi-Mensah et al (1995) also state that a system that meets its objectives is a successful system. These points could possibly suggest that once the requirements are met then there should be success. This suggests that the task should then be to find a way to improve the requirements that are received from users. It has been suggested that achieving success with Information Systems (IS) is a difficult task, particularly in the sense of meeting users needs and expectations (Whyte and Bytheway, 1995). However, IS success has also been measured by its ability to meet project expectations, monetary payback and the extend of usage of the latest technology etc (see chapter 2). The involvement of users has been discussed in some detail, showing some acknowledgement of the importance of their role in the systems development process. The industrial visits offered an opportunity to interact with users and developers. The sources used were reputable companies such as IBM, Logica, Ford and Rover. A larger sample would have produced more valid results. However, the size of the sample was determined by the response received from the letters that were sent out to individuals. Every positive response led to a meeting. Chapter 2 concluded that over the past 15 years the user has slowly become more visible in the systems development literature. This is also demonstrated by the introduction of RAD, where a group of users and the developer is placed in one room with continuous rapid development. This research suggests that it is important to achieve a balance between the responsibilities of the user and the developer, ensuring that their skills are used where they are most effective. The survey of users and developers (see chapter 4) has shown that users have shown a lack of trust in the systems. This suggests there is a problem with the results the systems are producing. The research also indicates that in order to achieve success, the original needs of the systems should be met (see chapter 2). This, therefore, suggests that the needs of users should be documented correctly. Therefore, the right people who hold the required information need to be involved. This can be seen more clearly in the problems that are associated with the lack of user involvement (see chapter 2 and 4). The literature suggests that considering the user affects worker productivity, worker satisfaction, worker motivation and job extent positively and can also be negative. The main problems are associated with the time barriers being extended, leading to projects running over budget.

Objective 3

Identify the requirements for a new method of capturing user requirements from both parties (users and developers).

This objective enabled the research conclusions to add an industrial perspective to the final method. The results suggest that the users of the systems are the people who have the information. Therefore, in order for the correct information to be in the systems, these people need to be involved to a greater extent.

A survey (see chapter 4) was conducted, in which users and developers from companies including BICC General Cables, Ford, Rover, IBM, Logica etc were interviewed. The interviews were carried out using a structured questionnaire where each user and developer was asked the same questions. The results were analysed using a clustering approach where groups of questions were clustered to draw conclusions about the information collected. This approach enabled first hand information to be collected from the participants. However, a number of different industries were used. This meant that detailed conclusions about any one particular industry could not be made. However, the diversity of the industries involved provided different industries with an opportunity to learn from each other.

A variety of requirements, such as involve all users at some stage, use groups and try to appoint a representative for the user group were found. These suggestions are limited to the experience of the users and the developers. Users have also shown an interest in the process, stating that they would be more willing to use the system if they were involved in the process more.

Objective 4

Develop a method for capturing user requirements. The method should consider the strengths offered by existing methods and the requirements stated by users and developers in industry. Then evaluate the method and identify what the results of the research indicate.

URCAM (User Requirements Capturing Method) was developed and then redeveloped after the initial evaluation according to the requirements of industry. The results of the evaluation are shown in chapter 6. URCAM has been able to address the lack of attention that existing methods have placed on capturing user requirements.

URCAM was evaluated in sections, due to problems with access to users and developers within the sponsoring company (BICC General Cables). This was due to the company (BICC General Cables) being taken over, which resulted in unforeseen changes. Therefore, the results obtained from the limited access to the sponsoring company were enhanced by contacting more companies. This proved successful, since four more companies were happy to get involved in the project. The time given by each of the companies was limited due to their own schedules, but it was sufficient to let various users and developers evaluate URCAM.

The initial evaluation showed that URCAM was too complicated (see chapter 6, section 6.2). The forms used in URCAM were also adjusted (see chapter 6, section 6.2). This led to a simplification of the forms. The final results from the initial evaluation suggested that the process of capturing user requirements should consist of a combination of users ticking boxes, filling in forms, and being interviewed verbally. This meant that users did not feel too much pressure during the form filling process, together with minimising the verbal interaction due to problems with misinterpretations (see chapter 2).

The main evaluation was conducted with the revised version of URCAM (see chapter 5 for revised version). The results indicted that using URCAM would produce higher quality results than previously used methods with the participating companies. The main areas where improvements were suggested are with the improvement in involvement that users would feel, an increase in the quality of information received from the users and the increase in usage of the new system that would be developed using URCAM. However, the evaluation did suggest that users would have some problems with filling in the URCAM forms. Possible reasons for these fears were that the users were apprehensive about being quoted, or that they might need to justify what they have said. However, a majority of the participants suggested that users would either have no fear or be cautious about filling in the forms (see table 6.2). The experts that participated in the evaluation of URCAM suggested that there were other considerations to improve URCAM. These were that back-up procedures need to be implemented and there was a need for flexibility (see chapter 6). However these points did not out-weigh the potential benefits suggested by the experts. These benefits included; things written down are reproduced better over the long term than verbal, opportunities for people doing the same sort of work to enable group support, enables time for thinking, able to review documentation and also presents a good opportunity to trace the information that has been received. However, these results are indications of what URCAM might achieve based on the experts experience. The ideal situation would be to test URCAM and then monitor the strengths and or weaknesses of the method after the implementation of a system. However, this was not feasible due to limitations mentioned earlier.

There are a number of specific points that the research indicates. These are as follows:

User involvement

According to the literature user involvement has been a major cause for concern in relation to systems success and failure. The increase in user involvement in systems development is likely to have a positive effect on the systems success and the company that it is being implemented in. Once this is increased the research indicates that there would be an increase in user satisfaction. Once there is satisfaction there is likely to be an increase in motivation and quality of inputs that the user gives to the project. This should eventually lead to an increase in systems success. However, once the user is involved it is likely to prolong the development process. In the past, developers have taken more control of the process and have only involved users when there is a specific requirement. With the increase in involvement, the user has an opportunity to be more aware of what is happening. Therefore, their expectations can increase, which should give them more control.

Hands on interaction

Users fill in forms during the URCAM process. They enable users to write down their present and future requirements and related information that is required as stated in the questions on the forms (see appendix D). The forms can also be used as templates for developers in the questioning process. This leads to an eventual sign off where users are able to state whether they agree with the specification and then agree with the final system that is produced.

The research indicates that by users filling in the forms themselves there is a feeling of more pressure on them to write down the correct information. This was tested in chapter 6. The feeling from the experts was that the user is likely to give higher quality information. Filling in the forms also made users feel more involved in the systems development process. This allows users to state exactly what they want, which again takes more control away from the developer. However, this also suggests that the developer should try to take more care in listening to the user and try to understand what they want and need.

Feedback from developers to users

Feedback is an important consideration (as stated by users within the survey of users and developers, see chapter 4). Feedback is carried out to let users know what is happening throughout the development process and to allow them to add their requirements as the development is taking place. This feedback requires developers to carry out the development of working programs whenever they can so that users are able to interact with what has been produced during the development process rather than at the end. Adequate feedback is required to ensure that the systems users and developers have the same goals/aims throughout the process.

During the survey of users and developers in chapter 4, users suggested that they require more feedback. Once there is feedback from the developer to the user, the research indicates that the user will be aware of what is happening. This also gives the user the opportunity to input constructive ideas about how things could be improved. This is likely to increase motivation, as they could feel more involved. The user is also likely to feel that what they have said is going to be taken into account in the development of the new system. However, this is also likely to require more commitment from the developer to let users know what is happening.

Structured approach to capturing user requirements

URCAM has adopted the structured approach (see chapter 2) and has offered the developer a set of phases and stages that they can follow. This approach does not have to be followed strictly without any deviation or adjustments. The developer should use the URCAM phases and stages as a template for them to guide themselves through the requirements capturing stage of the systems development process.

The structured approach has the ability to assist the developer (especially less experienced developers). This approach also gives the user the opportunity to remember the process. This possibly leads to the user being able to carry out more of the process without the developer (not too much, since the developers experience is important in the process). However, it is also possible (especially with more experienced developers) for them to resist the use of a structured approach due to them feeling tied down to one approach. This is likely to occur if the approach is not flexible. The literature suggests that a structured approach is more likely to assist a less experienced developer and act more like a guide to more experienced developers (see chapter 2).

The structured approach could possibly encourage users to get to know the process, which should lead to less training in the long term. The approach can also be improved as requirements of the method change and new methods are developed, offering positive stages that fit with the aims of URCAM.

Group interaction between developers and users

Various users from different departments or processes and developers should be involved at the same time. This group interaction should assist in the ideas being generated and agreed collectively (e.g. users and developers agreeing). This could lead to the interactive process producing worthwhile results for all parties involved. For example, if users suggest a function that is not feasible, the developer should be able to justify why it is not feasible rather than the user expecting to see the function and not seeing it at the end of the project.

During the survey of users and developers in chapter 4, users suggested that they would prefer the process of capturing user requirements to be completed in groups. Therefore, this was implemented in URCAM. The group situation was then evaluated in chapter 6. Users were placed in groups to test how they would use URCAM. The results suggested that the group situation increased the confidence of the users. It also allowed them to ask each other questions to help justify their answers. This group situation also helped the user and developer to communicate more. However, the results also suggested that it is not easy to form the groups, especially if there is not full buy-in from senior management. This problem occurred as a result of users not having enough free time away from their jobs to carry out the process. This in turn makes it more difficult to test the group situation. There are other factors to test such as group size and the social dynamics that occur when groups are formed. For example, whether some users are intimidated by the group situations or they say what others say because they do not want to be in a minority and they would rather go with the majority. These were not tested, since they were not within the aims and objectives of the study

Increased users and developers understanding of each others needs

In the past there has been a lack of interaction and understanding between developers and users in systems development (see chapter 2). This could be a result of developers feeling that they understand the needs of users and users are not really aware of what they require (see chapter 4). This suggests that one of the reasons why developers have produced inefficient systems in the past, is that users are not using them effectively. However, this problem is slowly being addressed and is addressed in the URCAM process. Once the users and developers are dependent on each other, then they need to understand each other in order to produce the desired results.

Once users and developers understand each other, the research indicates that there would be fewer problems with wrong interpretations of information given. This is in contrast to previous methods where the user answers questions verbally. Section 2.3.2 gives examples of the problems of verbal communications, where misinterpretation is a common problem. There is also a need to reduce the technical jargon (language barrier) that developers use. It has been suggested that this could make users feel uncomfortable (see chapter 1). Once the technical jargon is reduced the research suggests that users are more likely to understand what is happening and can therefore feel more comfortable getting involved in the process. Once the understanding is increased the research indicates that the requirements will also be improved. However, the major problems that may occur here are that it can take time for the level of understanding to reach this point. It is also possible that during this time problems such as wrong requirements are stated. This should be a short term problem, since increased communication should lead to increased understanding. This process can highlight more problems that take a long time to solve; for example developers may not want the user to understand what is happening. This is suggested due to the problems in communication still being present today after so many years of this problem being apparent in the literature. However, once the understanding has reached the desired level, the research suggests that future projects should have a better chance of success.

Increased quality of user input

URCAM was evaluated in chapter 6. During the evaluation users filled in the URCAM forms. The users and developers were also asked whether the quality of information given by users while filling in the forms is likely to be higher (if forms are filled in) than if verbal communication was used. The results suggested that the quality would be higher. The research indicates that the process of filling in the URCAM forms and generally being involved leads to the user taking more time and care with what they write down or say during the requirements capturing process. One possible result of this is higher quality inputs into the requirements specification document. This also gives the user the opportunity to demand more, which may be better, since the system may be more efficient. Alternatively, it may cause the project to last longer than expected.

More useful and trustworthy information in the system and the organisation

Once the output of the system is correct, the level of accuracy of the information is likely to be higher. This should lead to users using the information more. This should also mean that in the long term there would be a higher degree of trust in the information that is produced by the system. This is likely to lead to the information being used for the benefit of the organisation for forecasting and analytical purposes.

Once users put more thought into filling in the URCAM forms and the requirements capturing process is carried out more efficiently, there is a possibility that the quality of the information circulating around the system should increased (see chapter 6). The more the system does what it is supposed to do and produces the results that it is supposed to produce, the higher the likelihood that the systems will be used by the users. This suggests that there would be more trust in the system. Alternatively, however, (especially in the short term) it can lead to complacency. This means that developers may leave the systems for longer periods. The research suggests that the systems will be used more, indicating that they would have more confidence in the information produced. Once users' confidence in the information increases, it is probable that checks will need to be made more regularly by the developers. This can result in more pressure being placed on developers, since the expectations for higher quality information from users and the organisations is likely to increase.

Increased usage of the system

The survey of the users and developers suggested that users would be more willing to use the system once they were involved in the process of capturing user requirements. With an increase in usage, this should result in more information becoming available within the organisation to initiate improvements and increase control.

7.2 Application and potential users of URCAM

Being developed with inputs from a number of industrial sectors, URCAM can be adopted for various sectors. It has also had inputs from literature that are directed at manufacturing and general information systems development. This indicates that URCAM provides the user a greater degree of flexibility in the industry in which they operate. Flexibility is also provided in the form of the questions that developers can ask users. The questions can be changed according to the requirements of the specific project. The forms can be used as templates where the format of the forms can be changed according to the type of information required. The user should be filling in the forms. This gives them a chance to document their requirements. This should decrease the ease that developers could have had in the past where only verbal communication is used and developers are then able to adjust requirements to suit their own timescales and budgets. The user should now have more control over what is implemented in the system, and so should become more enthusiastic about the whole process. Various limitations, for example time allocated for carrying out the requirements capturing process, means that all of the stages of URCAM may not be completed. The most important stage, then, would be the use of the forms where the requirements are captured from the user. However, the ideal situation would be to go through all of the stages. The environment and number of participants can also be adjusted. This will depend on the location and the availability of the required users.

URCAM proposes a method for capturing user requirements. It can also be used to perform a general requirements capturing process. For example, identifying the raw materials required to produce a specific product. The forms are designed in such a way that information about a specific item can be listed and then further questions can be asked according to the information requirements of the project.

The developers most likely to see the benefits of using URCAM are as follows:

- Those who have used a method and have not been happy with the results and are looking for a new approach that will help them to improve their current working practices.

- Developers who are responsible for identifying the present systems requirements and need to analyse the information by asking why and how else questions in order to develop a proposed system.

- Companies who are finding that users are not giving correct information due to a lack of motivation while being involved in the process of capturing user requirements.

- Developers who want to understand a particular process within an organisation, identifying who sends and/or receives data to where so that all of the participants in the process can be identified.

- Inexperienced developers who would like to have some guidance in the form of a structure with stages and phases to complete.

- Developers without the necessary skills to analyse information and understand whether a requirement is needed or not.

7.3 Contribution

This research has made two main contributions. These are (i) a review of existing methods and industry, highlighting their strengths and weaknesses in relation to capturing user requirements and (ii) an amalgamation of the strengths offered by existing methods and industry to develop a new approach to capturing user requirements called URCAM. Each of these contributions are explained in more detail below:

(i) Review of existing methods and industry, highlighting their strengths and weaknesses in relation to capturing user requirements

As stated in chapter 2 these methods lack the user-developer interactive process. RAD offered the closest physical option and ETHICS offered the most social consideration. However, this contribution aimed to improve the users responsibility in the development process and get the user to work harder to achieve the aims of the company together with increasing their understanding of the expectations of the system and the needs of the developer. The research then looked for methods that claimed to understand the requirements specification stage; these also lacked the social side putting more emphasis on the technical issues with insufficient consideration of the needs of the user. This generated a generic picture of the aims of the methods, and realised that they were not dealing with how to get the requirements from the user; they concentrated on what the requirements should look like, how they should be represented, and who should be able to understand them. This did bring the users closer, but the research still saw that it was possible to get the user even more involved in the process, together with offering a structure for the process to be carried out. It was anticipated that a new method would achieve an increase in user involvement and therefore improve the effectiveness of the results of the systems development exercise.

(ii) Amalgamation of the strengths from existing methods and from industry to develop a new approach called URCAM

URCAM is aimed at improving the methods used for developing systems by adding a structured approach to the process of capturing user requirements. It offers an approach for developers to use as a guideline for interacting with the user to form a user requirements specification document. Various methods have been used by companies in the past to develop systems. Some companies may use methods from the beginning to the end, some have tailored a method to suit their needs, while others have developed their own in–house methods. URCAM has the flexibility to be changed according to the needs of various organisations.

The main areas where there is a clear difference between the existing methods and URCAM is firstly within the URCAM specification. It consists of attributes including (1) A structured approach, (2) Active user involvement, (3) Teamwork, (4) Adequate feedback and (5) Graphical representation. The existing methods offer these factors collectively, where URCAM offers these within an individual method as shown in figure 5.2. Secondly, URCAM enables the user to have more control with the use of the URCAM forms, where users state their requirements. Thirdly a structured approach is available. This approach aims to ensure that the user understands the process and feels as if their input is useful and will be considered when the system is being produced. The developer can also benefit from the structured approach offering a guide to follow throughout the process of capturing user requirements.

7.4 Limitations

This section identifies the weaknesses within the research methodology and the findings presented in the thesis.

As in any research, there would be a wish that more cases were used in order to produce conclusions based on the largest possible sample of companies that carry out systems development projects. In addition, more testing of URCAM would be beneficial to evaluate the method to justify further the need for a new approach and the proposed benefits of the new approach.

URCAM is based on literature and information acquired from various industrial sectors. However, although this research does provide a view from each sector involved, it does not provide a thorough view of each sectors' requirements. This also means that the industrial sectors/disciplines not included, may not see the benefits immediately, since their sectors were not involved in the investigation.

There are a number of methodologies, techniques and tools that could have been analysed, criticised and learned from. All of the above were not reviewed, since the time and resources available meant that it was not be feasible to identify all of them,. However, the major ones were reviewed. This suggests that the results gained are based only on the information received. Therefore, assumptions based on all methodologies, tools and techniques are not made.

Due to the lack of research in this particular subject area, the research methodology is based on programmes from work carried out in various areas of research; for example manufacturing systems development, information systems development and social research. However, the results achieved from the research determine whether the research methodology produced valid conclusions.

Time also proved to be a limiting factor. Companies who were involved in the cases offered a limited amount of time to interview users and developers for the survey (see chapter 4) and the evaluation (see chapter 6) of URCAM. This allowed the investigation to draw some conclusions. However, a greater degree of assistance would have presented the study with more valid conclusions, especially in the evaluation process although the time given was appreciated and useful.

7.5 Recommendations for further research

There are a number of areas that have been identified as areas that could increase the understanding of user involvement and capturing user requirements. Therefore, they have been included in this section to document ideas for further research.

The time that it should take to complete the process of capturing user requirements could be investigated. This presents a further opportunity for developers and the organisation to be able to predict how long a requirements capturing exercise should take. This is likely to assist in budgeting the total cost of the project and also to attempt to ensure that the requirements capturing stage is given adequate time to be completed successfully.

URCAM has been evaluated during the research as described in chapter 6. There is room for improvement. One idea for improvement would be that the various sections of the URCAM forms, where users input data could be extracted or copied onto paper and stuck to a wall (similar to mind mapping exercises). This would allow a collection of users and developers to move data flows around and show the sequences of events as they presently are and then show the proposed flow of data.

The development of a software application would also be a recommendation for future work. URCAM has the opportunity to extract a great amount of data that is contained within a system. The development of a software application would be able to store the data more effectively. This would also enable a graphical representation of the system to be shown.

Specific sectors can also benefit from URCAM. A detailed investigation into a sector or a specific company would ensure that their needs are met within the method. This would take into account the time required, the cost, the technical requirements and the people who would be involved in the process. This would require a thorough understanding of the company or individual sectors, together with thorough testing of the method produced.

The research has identified a need for a new method of capturing user requirements. In order to implement such a method there is likely to be a change in the way that things have been done in the past. A suggestion for further research would be to develop a change strategy so that the method could be implemented smoothly.

7.6 Conclusion

The final conclusions to the thesis suggest that there is a trend in systems being produced that are not successful. This means they are not giving the companies the results that were originally expected. One of the major reasons for this failure has been the lack of involvement of users during the development process. Many methodologies have been produced and cited, some concentrating on the whole life cycle others concentrating more on user requirements. However, after reviewing these methods the results suggest that there is still room for improvement. The main area where improvements could be made is in terms of their ability to recognise the importance of user involvement during the systems development process.

As discussed above, a solution to the problems associated with the user/developer relationship would come in the form of URCAM. URCAM has been produced taking into consideration a number of factors from a collection of existing methods, together with requirements from users and developers from industry. URCAM aims to improve the involvement of users by considering a number of factors such as 1) Hands on interaction where forms are filled in by users. 2) Feedback were developers produce prototypes and offer regular feedback to users about what they consider to be the requirements of the user. 3) A structured approach to capturing requirements where a developer has a structure to follow rather than changing the way requirements are collected, or using an ad hoc approach each time a systems development project is completed. 4) Group interaction where groups of users are involved at the same time with the developer. This group situation allows users to assist each other and it presents an opportunity for developers to get a closer working relationship with the systems users. These factors mentioned above should help the users and developers understand each others needs. This should eventually help developers to produce a system that the user has asked for.

Once the appropriate methods are used, the research suggests that users would feel that their involvement is required, which should lead to users having a higher degree of motivation and willingness to get involved. This should result in an increase in the quality of the inputs that users contribute to the project, which would mean that more useful and trustworthy information would be in the requirements specification. This should eventually lead to an increase in the usage of the system due to the user being involved more and feeling that their input is important to the success of the system.

URCAM has been evaluated and the results suggest that it has been successful in providing a solution that addresses the problem of the lack of user involvement during the process of capturing user requirements.

The main contributions that this thesis has made to the body of knowledge are firstly with its review of existing methods form the literature, where strengths and weaknesses have been identified that relate to the involvement of users within the systems development process. The weaknesses have been used as reasons why there is a need for a new approach and the strengths have been used to develop a specification for the new approach. Many factors from the literature were considered, including identifying the As-Is and the Should-Be, thinking about the social perspective, speed in the development process, identifying the origins of information, using data flow diagrams, analysing requirements, formalising the documentation process etc. Secondly, the factors from the literature were then added to those found during the survey of users and developers, these included suggestions from users and developers such as using guidelines, increase support, more feedback etc. These factors from the literature and the survey were amalgamated to develop a new method to carry out the process of capturing user requirements called URCAM.

The major difference between URCAM and other methods is that it offers a well defined structure for capturing user requirements that begins before the developer meets with the user, encouraging a thought process of the things that are to be achieved at the end of the exercise. The method gives a structure to follow that considers the needs of the user and the developer. It also presents a method that lets users fill in forms stating their present and future needs. These forms contain questions that will be defined according to the needs of the project. This is combined with a method of analysing what they currently have and what they would like by asking further questions, such as, why do you need the data? The method has been simplified to allow a less experienced IT user to understand what is happening and encourages both parties to try and understand each other better.

REFERENCES

1. Alavi, M. (1990) 'An empirical investigation of two alternative approaches to control of the end-user application development process'. Data Base, Winter Vol. 20, No. 4.

2. Alavi, M. and Weiss, I. (1985) 'Managing the risks of end-user computing'. Journal of MIS, Vol. 11.

3. Aldenderfer, M. and Blashfield, R. (1984) *Cluster Analysis.* Sage University Papers.

4. Alford, M. (1977) 'A requirements engineering methodology for real time processing requirements'. IEEE Trans Software Engineering, Vol. SE-3, pp 60-69. Jan.

5. Allen, P. and Frost, S. (1998) *Component-Based Development for Enterprise Systems. Applying the SELECT perspective.* Cambridge University Press.

6. Alter, S. (1996) *Information Systems: a management perspective.* 2nd ed. Menlo Park, CA: The Benjamin/Cummings Publishing Company.

7. Amoako-Gyampah, K. and White, K. (1993) 'User involvement and user satisfaction- an exploratory contingency model'. Information and Management, Vol. 25, No 1 July.

8. Anastasi, A. (1990) *Psychological testing.* 6th edn. Collier-Macmillan.

9. Avison, D. E. and Fitzgerald, G. (1995) *Information System Development: Methodologies, Techniques and tools.* Second edition. McGraw Hill 1995.

10. Babin. G., Lustman, F. and Shoval, P. (1991) 'Specification and Design of Transactions in Information Systems: A formal Approach'. IEEE Transactions on Software Engineering, Vol. 17, No. 8.

11. Bailey, K. (1987) *Methods of social research.* The Free Press.

12. Ball, G. H. (1971) *Classification Analysis.* Standard research Institute SRI Project 5533.

13. Ballantine, J. A., Galliers, R.D. and Stray, S.J. (1993) Information Strategy/Technology Investment decisions: *The Use of capital Appraisal Techniques in Organisations. Proceedings of the International Conference on IT Evaluation*, Henley, July.

14. Barki, H. and Hartwick, J. (1989) 'Rethinking the concept of user involvement'. MIS Quarterly, Vol. 13, (1). March, pp 53-63.

15. Barki, H. and Hartwick, J. (1994) 'Measuring user participation, user involvement and user attitudes'. MIS Quarterly, Vol. 18, (1), March.

16. Bell, J. (1987) *Doing your research project. A guide for first–time researchers in Education and Social Science.* Open University Press.

17. Benjamin, R. I. (1982) 'Information technology in the 1900's: A long range planning scenario'. MIS Quarterly, Vol. 6, (2), June, pp 11-31.

18. Bekker, M. and Vermeeren, A. (1993*) Developing user interface design tools: An analysis of interface design practice.* In Loresey, E (ed) Contemporary Ergonomics 1993, London: Taylor and Francis pp 79-84.

19. Benson, D.H. (1983) 'A field study of End-User Computing: Findings and issues'. MIS Quarterly, Vol. 7, pp 35-46.

20. Berdie, D., Anderson, J. and Niebuhr, M. (1986) *Questionnaires: Design and Use*. Second edition. The scarecrow Press, Inc.

21. Bernus, P. and Nemes, L. (1996) 'A framework to define a generic enterprise reference architecture and methodology'. Computer Integrated Manufacturing Systems, Vol. 9, No. 3, pp 179-191.

22. Berelson, B. (1971) *Content analysis in communications research*. New York: Hafner.

23. Bigler, M. (1995) 'Developing end-user computing guidelines'. EOPACS newsletter, Vol. 22, No. 12, June, pp. 1-5.

24. Black, T. (1993) *Evaluating Social Science Research. An Introduction*. Sage Publications.

25. Blalock, H. (1960) *Social Statistics*. London: McGraw Hill.

26. Boehm, B. and Ross, R. (1989) 'Theory W software project management. Principles and examples'. IEEE Transactions on Software Engineering, Vol. 15, pp 902-15.

27. Booch, G. (1991) *Object Oriented Design with applications*. Benjamin/Cummings Redwood City California.

28. Booch, G. (1994) *Object Oriented Analysis and Design with applications*. Second edition. Benjamin/Cummings.

29. Brooks, F. (1987) 'No Silver bullet: Essence and accidents of software engineering'. Computer Journal, Vol. 19, No. 4, pp 10-19, April.

30. Brooks, L. and Jones, M. (1996) *CSCW and requirements analysis: Requirements as co-operation/requirements for co-operation.* CSCW Requirements and Evaluation. Springer BCS, pp 10-20.

31. Bubenko, J Jr. (1986) *Information systems methodologies – A research view, In: Information Systems Design Methodologies: Improving the practice.* Elsiver Science Publications, North Holland.

32. Burrows, P. (1994) 'Compaq: ready to Rumble'. Business Week, Vol. 29, August 22.

33. Busby, J. and Williams, G. M. (1993) 'The value and limitations of using process models to describe the manufacturing organisation'. International Journal of Production research, Vol. 31, No 9, September, pp 2179-2194.

34. Bytheway, A. (1994) *Business Modelling Lecture Notes.* SOM, Cranfield University.

35. Byrne, J. and Barlow, T. (1993) Structured Brainstorming: A method for collecting user requirements. *Proceedings of the Human factor and Ergonomics Society 37th Annual Meeting.*

36. Caginalp, E. G. (1994) 'Small-to-Midsize Companies Light Fire Under PC Sales'. Computer Reseller News, Vol. 2, August 29.

37. Cale, E. G. and Curley, K. F. (1987) 'Measuring Implementation outcome: beyond success and failure'. Information and Management, Vol. 13, No. 5.

38. Cantamessa, M. and Paolucci, E. (1998) 'Using organisational analysis and IDEF0 for enterprise modelling in SMEs'. International Journal of Computer Integrated Manufacturing. Vol. 11, No. 5, pp 416-429.

39. Carley, K. (1990) *Content analysis*. In R. E. Asher et al (Eds.) The encyclopaedia of language and linguistics. Elmsford, NY: Pergamon.

40. Carnall, C. (1986) 'Managing strategic change: an integrated approach'. Long Range Planning, Vol. 19, pp 105-15.

41. Carnier, D. T. (1958) 'And how to avoid them'. Computer journal, Vol. 1.

42. Carroll, J., Rosson, M., Chin, G. and Koenemann, J. (1998) 'Requirements Development in Scenario-Based Design'. IEEE Transactions on Software Engineering, Vol. 24, No. 12, Dec.

43. Carter, J., Whytes, I., Birchall, S. and Swatman, D. (1997) 'Rapid Integration and Testing of business solutions'. BT Technology journal, Vol. 13, No. 3, July.

44. Chen, D., Vallespir, B. and Doumeingts, G. (1997) 'GRAI integrated methodology and its mapping onto generic enterprise reference architecture and methodology'. Computers in industry, Vol. 33, No. 2,3, pp 387-394.

45. Chen, P. P. S. (1976) 'The Entity Relationship Approach: Towards a Unified View of Data'. ACM Trans. Database Systems, Vol. 1, No. 1, pp 9-36, March.

46. Cheney, P. and Dickson, G. (1982) 'Organisational Characteristics and IS: An exploratory investigation'. Academy and Management journal, March.

47. Clarke, R. and Moreira, A. (1999) 'Formal specification of user requirements'. Automated software engineering, Vol. 6, pp 217-232.

48. Clavadetscer, C. (1998) 'User involvement key to success'. IEEE Software, Vol. 15, Issue 2, March –April.

49. Coad, P and Yourdon, E. (1991) *Object Oriented Analysis*. Second edition. Prentice Hall, Englewood Cliffs, New Jersey.

50. Cohen, L. and Manion, L. (1989) *Research Methods in Education.* 3rd edn. London: Routledge.

51. Colter, M. A. (1984) 'A Comparative Examination of Systems Analysis Techniques'. MIS Quarterly, Vol. 8, (1), March.

52. Corbett, J. M., Rasmussen, L. B. and Rauner, F. (1991) *Crossing the boarder: The social and engineering design of computer integrated manufacturing systems*. Springer-Verlag, London Corporate Source: Manchester Business Sch, Manchester, UK.

53. Cougar, J. D. and Colter, M. A. (1985) *Maintenance programming: Improving productivity through motivation.* Prentice-Hall, Englewood Cliffs, NJ.

54. Cronbach, L. J. (1990) *Essentials of psychological testing.* 5th edn. Harper and Row.

55. Culnan, J (1986) 'The intellectual development of management information systems. 1972-82: a co-citation analysis'. Management science, Vol. 32, No. 2, pp 156-172.

56. Curtis. B., Krasner, H. and Iscoe, N. (1988) 'A field study of the software design process for large systems'. Communications of the ACM, Vol. 31, No. 11, pp 249-266.

57. Czaja, R. and Blair, J. (1995) *Designing Surveys. A guide to decisions and procedures*. Pine Forge press.

58. Dale, B. G. (1994) *Managing Quality*. Prentice Hall international (UK) Ltd., Second edition, London, (Ch.27).

59. Darke, P. and Shanks, G. (1996) Viewpoint development for requirements definition: *Towards a conceptual framework. Proceedings of the 6th Australasian conference on information systems.* Curtin University of Technology, Perth.

60. Davis, F. (1959) 'The cabdriver and his fare: facets of a fleeting relationship'. American Journal of Sociology. Vol. 65, pp 158-165.

61. Davis, G. B. and Olson, M. H. (1985) *Management Information Systems. Conceptual foundations, Structure and Development.* McGraw-Hill, New York.

62. Davis, G. B. (1982) 'Strategies for information requirements determination'. IBM Systems Journal, Vol. 21, No. 1.

63. Dean, D., Orwig, R., Lee, J. and Vagel, D. (1994) Methodology with a group modelling tool: *Group support, model quality and validation. In: Proceedings of the 27th Hawaii international conference and system science*, 4th-7th January, pp 214-223.

64. DeBrabander, B. and Edstorm, A. (1977) 'Successful Information Systems Development Projects'. Management Science, Vol. 24, (2), pp 191-99.

65. DeMarco, T. (1979) *Structured analysis and systems specification.* Prentice-Hall. Inc, Englewood Cliffs, NJ.

66. Dey, I. (1993) *Qualitative data analysis. A user-friendly guide for social scientists.* Routledge London and New York.

67. Doumeingts, G., Chen, D. and Vallespir, P. (1993) GIM (GRAI Integrated Methodology) and its evolution. A methodology to design and specify Advanced Manufacturing Systems. *Information Infrastructure Systems for Manufacturing. Proceedings of the workshop on the design of information infrastructure systems for manufacturing.* DIISM 1993, Tokyo, Japan 8-10 Nov 1993.

68. Doumeingts, G. (1985) 'How to decentralise decisions through GRAI model in production management'. Computers in Industry, Vol. 6, No. 6, 501-514.

69. Downs, E., Clarke, P. and Coe, I. (1992) *Structured systems analysis and design method: application and context.* Prentice-Hall International (UK) Hertfordshire.

70. DSDM Consortium. (1997) *Dynamic Systems Development Method (DSDM) Version 3.0.* Tesseract Publishing, September.

71. Dyer, H. and Bouchet, M. L. (1995) *A comparison between the perceived value of information retrieved via end-user searching of CD-ROMs and mediated online searching.* British Library (BLRD Report 6208).

72. Earl, H. (1978) 'Prototype-systems for accounting information and control'. Accounting Organisation and Society, Vol. 3, (2), pp 161-70.

73. Earl, M. J., Fenny, D., Lockett, M. and Runge, D. (1988) IT competitive advantage and innovation: maxims for senior managers. Research and Discussion papers, ROP 88/2. Oxford Institute of Information Management, Templeton College, Oxford.

74. Edberge, D. and Bowman, B. (1996) 'User developed applications: an empirical study of application quality and developer productivity'. Journal of Management Information Systems, Vol. 13, No. 1, Summer, pp 167-185.

75. EDP Analyzer. (1983) 'Future Effects of End-User Computing' Vol. 21, No. 11.

76. Edwards, H. M., Thompson, J. B. and Smith, P. (1989) 'Results of survey of use of SSADM in commercial and government sectors in the United Kingdom'. Information and Software Technology, Vol. 31, No. 31, Jan/Feb.

77. Edwards, C., Ward, J. and Bytheway, A. (1995) *The Essence of Information Systems*. Second edition. Prentice Hall.

78. Ellis, D. (1993) 'Modelling the information seeking patterns of academic researchers: a grounded theory approach'. The Library Quarterly, Vol. 63, (4), pp 470-484.

79. Engler, N. (1996) 'Bring in the user'. Computer world, Vol. 30.

80. Evans, R. (1989) 'Reaping the full benefits of end-user computing'. Computer world, Vol. 23.

81. Everit, B. (1980) *Cluster Analysis*. Halsted Press Second Edition.

82. Ewusi-Mensah, K. and Przanyski, R. (1995) 'Learning from abandoned Information Systems development'. Journal of Information technology, Vol. 10, No. 1, March, pp 3-14.

83. Fischer, K. and Walker, M. (1979) 'Improved software reliability through requirements verification'. IEEE Transactions *Rel.*, Vol. R-28, Aug, pp 233-240.

84. Fitzgerald, B. (1996) 'Formalised systems development methodologies a critical perspective'. Information Systems Journal, Vol. 6, Number 1, Jan.

85. Fitzgerald, B. (1997) 'The use of software development methodologies in practice: a field study'. Information Systems Journal, Vol. 7, No. 3, July, pp 201-212.

86. Fitz-Gibbon, C. and Morris, L. (1988) *How to design a program evaluation.* Sage publications Inc.

87. Flynn, D. and Warhurst, R. (1994) 'An empirical study of the validation process within systems development'. Information Systems Journal, Vol. 4, pp 185-212.

88. Folkes, S and Stubenvoll, S (1992) Accelerated systems development. Prentice hall. New York. 2nd ed.

89. Galliers, R. (1992) *Information Systems Research: Issues, Methods and Practical Guidelines.* Blackwell, Oxford.

90. Gasson, S. (1993) Design Processes in Information Systems Development Projects. Warwick Business School Research Bureau. No. 103.

91. Gavron, H. (1966) *The captive wife.* London, Routledge and Kegan Paul.

92. Gibson, C. and Nolan, R. (1974) Managing the fair stages of EDP growth. Harvard Business Review, January/February, pp 76-88.

93. Gilbert, T. E. (1978) *Human Competence, Engineering Worthy Performance.* McGraw Hill, New York.

94. Gorry, G. and Morton, M. (1971) 'A framework for management information systems'. Sloan Management review. Fall.

95. Gougen, J. and Linde, C. (1993) Techniques for requirements elicitation. *Proceedings of the first IEEE International Symposium on Requirements Engineering (RE '93)* San Diego, USA 4-6 Jan, pp 152-164.

96. Gough, P. A., Fodemski, F. T., Higgins, S. and Ray, S. J. (1995) "Scenarios-An industrial case Study and Hypermedia Enhancements". *Proc. Second IEEE Symp. Requirements Engineering,* IEEE Computer Society, pp 10-17.

97. Gough, M. (1997) 'RADical enhancements with "time to market" as the basis of competition, IT organisations that can master RAD gain an edge'. Application Development Advisor, Vol. 1, No. 1, Sept/Oct 1997.

98. Greenspan, S., Mylopoulos, J. and Borgida, A. (1994) On formal requirements modelling languages: *RML revisited. Proceedings of the 16th International Conference of Software Engineering,* IEEE Press, Los Alamitos.

99. Gremillion, L. L. and Pyburn, P. (1983) 'Breaking the systems development bottleneck'. Harvard Business Review, March-April.

100. Grindley, K. (1995) *Managing IT at Board Level.* Pitman.

101. Gundogan, M. (1995) *An implementation Model of a Quality Management Information Scheme for Cellular Manufacturing Environment.* Cranfield University Thesis.

102. Hanson, D. J. (1980) 'Relationship between Methods and judges in Attitude Behaviour Research'. Psychology, Vol. 17.

103. Haumer, P., Pohl, K. and Weidenhaupt, K. (1998) 'Requirements Elicitation and Validation with real world scenes'. IEEE Transactions on software engineering, Vol. 24, No 12, December.

104. Heichler, E. and Cafusso, R. (1995) 'Methodologies sought to solve client/server puzzle'. Computer world, Vol. 29, (6).

105. Henderson, J. C. and Treacy, M. E. (1984) "Managing End-User Computing" CISR Working Paper 120, Sloan School of Management, MIT October.

106. Heyink, J. and Tymstra, T. (1993) 'The function of qualitative research'. Social Indicators research, Vol. 29, (3), pp 291-305.

107. Hirschheim, R. (1985) *Office automation: a social organisational perspective*. Chichester: Wiley.

108. Holsti, O. R. (1968) *Content analysis*. In G. Lindzey and E. Aronson (Eds.) Handbook of social psychology: Vol. 2 research methods (2nd ed. Pp. 596-692) Reading, MA: Addison-Wesley.

109. Holsti, O. R. (1969) *Content analysis for the social sciences and the humanities*. Addison-Wesley: Reading, MA.

110. Hopker, O. (1994) *Evaluation and choice of information Systems Methodologies – A Welsh Perspective*. In: Information Systems Methodologies, Lissoni et al (eds) pp 93-103. The BCS Information Systems Methodologies Group, Heriot-Watt University, Edinburgh.

111. Howard, J. S. and Smith, R. D. (1986) 'Computer anxiety in management: Myth or reality?'. Communications of the ACM, Vol. 29, (7), July, pp 611-615.

112. Hsieh-Yee, I. (1990) *The search tactics of novice and experienced searchers: A comparative study*. University of Wisconsin-Madison.

113. Humphrey, W., Snyder, T. and Willis. R. (1991) 'Software process improvement at Hughes Aircraft'. IEEE Software, Vol. 8, No. 4, pp 11-23.

114. IBM. (1961) Study Organisation Plan Documentation Techniques. Manual C20-8075, pp 1-26.

115. Igbaria, M., Pavri, F. N and Huff, S. L. (1989) 'Microcomputer Applications: An Empirical look at usage'. Information and Management, Vol. 16, (4), April, pp 187-196.

116. Ince, M. (1996) *The integration of Information Systems. A case study for BICC.* Cranfield University Thesis.

117. Ince, M. and Kay, J. (1998) The development of a user-oriented information systems development strategy. *Advances in manufacturing technology. Proceedings of the 14th National Conference on Manufacturing Research.* University of Derby. UK 7-9 September.

118. Ives, B. and Olson, M. (1984) 'User involvement and MIS success: A review of research'. Management Science, Vol. 30, Issue 5, pp 586-604.

119. Jacobson, I., Christrson, M., Jonsson, P. and Overgaard, G. (1992) *Object-Oriented software engineering – A use case driven approach.* Addison-Wesley.

120. Jankowicz, A. D. (1995) *Business research projects.* 2nd ed. London: Chapman and Hall.

121. Jayaratna, N. (1994) *Understanding and Evaluating Methodologies: A Systematic framework.* McGraw Hill, London.

122. Kane, E. (1984) *Doing your own research.* London: Marion Boyars publishers Ltd.

123. Keen, J. (1987) *Managing systems development.* 2nd ed. John Wiley and Sons Ltd.

124. Kerlinger, F. N. (1986) *Foundation of Behavioural Research,* 3rd edn. Holt, Rinehart and Winston.

125. Keyes, J. (1992) 'How software is developed undergoing many changes'. Software magazine, 12 (1), 38-40.

126. Koh, I. S. J. and Heng, M. S. H. (1996) 'Users and designers as partners – design method and tools for user participation and designer accountability within the design process'. Information Systems Journal, Vol. 6, No. 4, October.

127. Komi, K. (1995) *A comparison of GRAI, SSADMv4 and IDEF0 as tools to analyse manufacturing systems.* Cranfield University Thesis.

128. Kosanke, K., Vernadata, F. and Zelm, M. (1997) *CIMOSA process Model for Enterprise Modelling.* Information Infrastructure systems for manufacturing. Elsevier Science, North-Holland, pp 59-68.

129. Kosanke, K. (1995) 'CIMOSA – Overview and status'. Computers in industry, Vol. 27, No. 2, pp 101-109.

130. Krippendorff, K. (1980) *Content analysis: An introduction to its methodology.* Beverley Hills, CA: Sage.

131. Kyng, M. (1991) 'Designing for co-operation: co-operating in design'. Communications of the ACM, Vol. 34, pp 65-73.

132. Kyng, M. (1995) Creating Contexts for Design. *Scenario Based Design: Envisioning Work and Technology in Systems Development.* New York: John Wiley & Sons, pp 85-107.

133. Lacity, M. and Hirschheim, R. (1993) *Information systems outsourcing: myths, metaphors and realities*. Chichester: John Wiley.

134. Land, F. (1982) 'Notes on participation'. Computer Journal, Vol. 25, No. 2, pp 40-45.

135. Langefors, B. (1963) 'Some Approaches to the theory of Information Systems'. BIT, Vol. 3, pp 229-254.

136. Law, D. and Longworth, G. (1987) *Systems development strategies and techniques*. NCC Publications. Manchester.

137. Lawler, E. and Hall, D. (1970) 'The relationship of job characteristics to job involvement, satisfaction and intrinsic motivation'. Journal of applied psychology, Vol. 54, (4), pp 305-312.

138. Lawrence, P. and Lorsch, J. (1967) *Organisation and Environment.* Harvard Press.

139. Lee, W., Cha, S. and Kwon, Y. (1998) 'Integration and analysis of use cases using modular petri nets in requirements engineering'. IEEE Transactions on software engineering, Vol. 24, No. 12, December.

140. Leedy, P. (1985) *Practical research: planning and design.* 3rd ed. New York: Macmillan.

141. Leonard-Barton, D (1987) 'Implementing structured software methodologies: a case of innovation'. Process Technology Interfaces, Vol. 17, No. 3, pp 6-17.

142. Leonard-Barton, D (1988) 'Implementation characteristics of organisational innovations'. Communication research, Vol. 15, pp 603-31.

143. Longworth, G. (1985) *Designing systems for change*. NCC, Manchester.

144. Longworth, G. (1992) *A users guide to SSADM version 4*. NCC, Blackwell Limited.

145. Lor, K. and Berry, D. (1991) 'Automatic Synthesis of SARA Design Models from System Requirements'. IEEE Transactions on software engineering, Vol. 17, No. 12, December 1991.

146. Lubars, M., Potts, C. and Richter, C. (1993) A review of the state of the practice in requirements modelling. In: *IEEE International Symposium on Requirements Engineering*, 1993, San Diego, California pp. 2-15 IEEE Computer Society Press, Los Alamitos, CA.

147. Lucas, H. (1992) *The analysis, design and implementation of information systems*. New York: McGraw-Hill.

148. Lynch, H (1969) 'ADS: A technique in systems documentation'. Data Base, Vol. 1, No. 1, Spring, pp 6-18.

149. Magal, S. R. (1991) 'A model for Evaluating Information Centre Success'. Management Information Systems, Vol. 8, No. 1, Summer.

150. Martin, J. (1991) *Rapid Application Development*. Macmillan Publishing Company.

151. McCarthy, I., Chodari, A. and Ridgway, K. (1994) A model of the production management systems in a make to stock company. In: *Advances in manufacturing technology VIII. Proceedings of the tenth national conference on manufacturing research, Loughborough University of technology,* 5-7 September 1994.

152. McDonough, A. (1969) *Centralised Systems Planning and Control.* Wayne, Pa.: Thompson Book Co.

153. McFarlan, F. and McKeeny, J. (1983) *Corporate information systems management: the issues facing senior executives.* Homewood, Ill: R.D. Irwin.

154. McGolpin, P. (1996) *An examination of the Inter-related factors and issues affecting the degree of success with Information Systems throughout the application lifecycle.* Cranfield University School Of Management Thesis.

155. Mehrens, W. A. and Lehmann, I. J. (1984*) Measuring and Evaluation in Education and Psychology*, 3rd edn. Holt-saunders.

156. Miles, M. and Hubermen, A. (1994) *An expanded sourcebook. Qualitative Data Analysis.* Second edition. Sage Publications.

157. Mitchell, P. (1997) 'Foundations Class. With so many projects ending in disaster, what hope is there for a company embarking on new projects?'. IBM Systems User, Vol. 18, No. 9, October.

158. Monk, P. (1991) Economics specification of user requirements. University of Strathclyde, working paper 91/6, May.

159. Moore, D. S. (1991) *Statistics: Concepts and Controversies.* 3rd ed. W.H. Freeman.

160. Morris, L. Fitz-Gibbon, C. and Lindheim, E. (1987) *How to measure performance and use tests*. Sage Publications Inc.

161. Morrison, D. G. (1967) 'Measurement problems in cluster analysis'. Management Scienc, Vol. 13, pp 775-780.

162. Moser. and Kalton. (1971) *Survey Methods in Social Investigations.* 2nd edn. Heinmann educational books.

163. Mumford, E. (1983) *Designing human systems*. Manchester Business school Press.

164. Mumford, E. (1994) Technology, communication and Freedom: *Is there a relationship. Proceedings of the IFIP WG8.2 working conference on Information Technology and new emergent forms of IFIP Transactions.*

165. Mumford, E. and Weir, M. (1979) *Computer Systems in work design-the ETHICS method.* Wiley, New York.

166. Mumford, E. (1997) 'The reality of participative systems design: contributing to stability in a rocking boat. Information'. Systems Journal, Vol. 7, No. 4, October, pp 309-322.

167. Mumford, E. and Hensall, D. (1978) 'A participative approach to the design of computer system'. Impact of Science on society, Vol. 28, No. 3, pp. 233-56.

168. Nantz, K. (1990) 'Supporting end-user applications development with the information transformation analysis management model'. Journal of microcomputer systems management, Vol. 2, No. 3, pp 9-15.

169. Nauman, J. and Jenkins, A. (1982) 'Prototyping: the new paradigm for systems development'. MIS Quarterly, Vol. 6, No. 3, pp 29-44.

170. Necco, C. R., Gordon, C. R. and Tsai, N. W. (1987) 'Systems analysis and design: current practices'. MIS quarterly, Vol. 11, No. 4, pp 461-76.

171. Newman, M. and Robey, D. (1992) 'A social process model of user-analyst relationships'. MIS Quarterly, Vol. 16, No. 2, pp 249-266.

172. Nielsen, P. A. (1989) 'Reflections on development methods for information systems: a set of distinctions between methods'. Office, Technology and People, Vol. 5, No. 2, pp 81-104.

173. Norman, D. (1988) *The psychology of everyday things*. Basic books.

174. Norton, D. and McFarlan, F. W. (1975) *The Information Systems Handbook*, Dow-Jones-Irwin, Homewood, IL, pp 517-528.

175. Olerup, A. (1991) 'Design approaches: a comparative study of information systems design and architectural design'. The Computer Journal. Vol. 34, pp 215-224.

176. Olson, M and Ives, B (1981) 'User involvement in systems design: An empirical test of alternative approaches', Information and Management, Vol. 4, Issue 4, September, pp 183-196.

177. Open University. (1994) *Software engineering*. Open University.

178. Oppenheim, A. N. (1992) *Questionnaire design, Interviewing and Attitude Measurement*. Pinter Publishers London and New York.

179. Orr, K. (1981) *Structured Requirements Definition*. Ken Orr and Associates. Inc, Topeka, KS.

180. Oskamp, S. (1977) *Methods of studying behaviour*. In L.S. Wrightsman (ed.), Social Psychology, 2nd edn. Monetery, Ca:Brooks/Cole. 191.

181. Page-Jones, M. (1991) 'Structured methods are dead. Long live structured methods'. American Programmer, Vol. 4, No. 2, pp 1-28.

182. Parker, N. M., Trainor, H. E. and Benson, R. J. (1989) *Information Strategy and Economics*. Prentice-Hall.

183. Parlett, M. and Hamilton, D. (1972) *Evaluation as Illumination.* Centre for Research in Educational Sciences, University of Edinburgh.

184. Patton, M. (1980) *Qualitative Evaluation Methods*, London: Sage.

185. Perry, W. E. (1991) *Quality Assurance for Information Systems*. QED Information Sciences. Inc. USA (Welleshey), (ch. 9-10).

186. Piaget, J. and Inhelder, B. (1969) *The Psychology of the Child.* New York: Basic Books.

187. Plihon, V., Ralyte, J., Benjamen, A., Maiden, N., Sutcliffe, D. and Heymans, P. (1998) A Reuse-Oriented Approach for the construction of scenario based methods. *Proc. Int'l Software Process Assoc. fifth Int'l Conf. Software process (ICSP '98),* Chicago, pp 14-17, June.

188. Palvia, P. and Nosek, J. (1993) 'A field examination of systems life cycle techniques and methodologies'. Information and Management, Vol. 25, No. 2, pp 73-84

189. Poulymenakou, A. and Holmes, A. (1996) 'A contingency framework for the investigation of information systems failure'. European journal of information systems, Vol. 5, No. 1, March, pp 34-47.

190. Potts, C. (1997) Fitness for use: *The System Quality that Matters most. Proc. Third Int'l Workshop requirements Eng.: Foundations of Software Quality REFSQ '97*, pp 15-28, Barcelona, June.

191. Pourmaleski, A. and Cardinali, R. (1994) 'Bridging the gap: human systems engineering'. International journal of Computer Applications in Technology, Vol. 5, No. 1/2.

192. Powney, J. and Watts, M. (1987) *Interviewing in Education research.* London: Routledge and Kegan Paul.

193. Purvis, R. and Sambamurthy, B. (1997) 'An examination of designer and user perceptions of JAD and the traditional IS design methodology'. Information and Management, Vol. 32, pp 123-135.

194. Qurashi, Z., Biss, T. and Polkinghorne, N. (1995) Design of a manufacturing based management information system. In: *Proceedings of the eleventh national conference on manufacturing research* (Advances in manufacturing technolgy IX Editied by Stockton, D and Wainwright, C). Taylor and Francis Ltd., London, pp 499-503.

195. Reubenstein, H. (1991) 'The requirements apprentice: automated assistance for requirements acquisition'. IEEE transactions on software engineering, Vol. 17, No. 3, March.

196. Ridgway, K. (1995) The GRAI method. Lecture notes, University of Sheffield.

197. Rittel, H. (1984) *Second-generation design methods.* In: Developments in design methodology, Cross, N(ed), John Wiley and Sons UK.

198. Rivard, S. and Huff, S. L. (1988) 'Factors of success of End-User computing'. Communications of the ACM, Vol. 31, No. 5, May.

199. Robey, D. and Marks, W. (1984) Rivals in information systems design. MIS Quarterly, Vol. 10, No. 1, pp. 5-15.

200. Robson, C. (1994) *Real World Research. A Resource for social scientists and Practitioner-Researchers.* Blackwell.

201. Roche, S. (1998) 'Dance to the music', Computing Journal, 11 June.

202. Roger, T. (1995) "Using Scenario-Based Designs to review user interface changes and enhancements" *Proc. DIS95:Designing Interactive Systems*, Ann Arbor, pp 237-246.

203. Rolland, C., Souveyet, C. and Achour, C. (1998) 'Guiding goal modelling using scenarios'. IEEE Transactions on software engineering, Vol. 24, No. 12, December.

204. Ruddock, R. (1981) *Evaluation: A Consideration of principles and methods.* Manchester Monographs.

205. Russo, N. L., Wynekoop, J. L. and Walz, D. B. (1995) *The use and adaptation of systems development methodologies. In Managing information and communication in a changing global environment.* Idea Group Publishing, Harrisburg PA.

206. Rutman, L. (1977) *Planning an evaluation. Evaluation Research Methods: A basic guide.* Sage Publications Beverly Hills London.

207. Shah, H., Dingley, S. and Golder, P. (1995) 'Bridging the culture gap between users and developers'. Journal of Systems Management, pp 18-21.

208. Sherif, C. W., Sherif, M. and Nebergall, L. R. (1965) *Attitude and attitude change*. Sanders, Philadelphia, PA.

209. Shoval, P. (1988) 'ADISSA: architectural design of information systems based on structural analysis'. Information Systems, Vol. 13, pp 193-210.

210. Somogyi, E. and Galliers, R. (1987) 'From data processing to strategic information systems. A historical perspective'. Towards strategic information systems. Vol. 1 Strategic information systems management series. Abacus Press. London, pp 3-25 .

211. Southwell, K. (1993) *Managing Software Engineering Teams.* In: Software Engineering Reference book, McDermid (ed) Butterworth-Heinmann.

212. Sprague, R. and McNurlin, B. (1993) *Information Systems Management in practice*. Englewood Cliffs London. Prentice Hall.

213. Stapleton, J. (1997) *DSDM Dynamic Systems Development Method. The method in practice*. Addison – Wesley.

214. Straub, W., Ang, S. and Evaristo, R. (1994) 'Normative standards for IS research'. Data base, Vol. 25, No. 1, pp 21-33.

215. Sutcliffe, A., Maiden, N., Minocha, S. and Manuel, D. (1998) 'Supporting Scenario-Based Requirements Engineering'. IEEE Transactions on Software Engineering, Vol. 24, No. 12, December.

216. Tait, P. and Vessey, I. (1988) 'The effect of user involvement on systems success: A contingency Approach'. MIS Quarterly, Vol. 12, No. 1, March, pp 91-108.

217. Taylor, M. J., Moynihan, E. P. and Wood-Harper, A. T. (1998) 'End-User Computing and Information systems methodologies'. Information Systems Journal, Vol. 8, No. 1, January, pp 85-96

218. Teichrow, D. and Hershey, E. III. (1977) 'PSL/PSA: a computer-aided technique for structure documentation and analysis of information processing system'. IEEE Transactions on Software Engineering, Vol. SE-3, pp 41-48, Jan.

219. Toh, K., Newman, S. and Bell, R. (1997) An architecture for information specification based on an SME model. *Proceedings of the 5th International Conference on FACTORY 2000*, 2-4 April 1997. Conference publication No. 435. IEE 1997.

220. Thomas, P. (1996) *CSCW requirements and evaluation.* London: Springer-Verlag.

221. Tourangeau, R. and Rasinski, K. (1988) Cognitive Processes Underlying Context Effects in Attitude Measurement, Psychological Bulletin, 103:299-314.

222. Treacy, M. E. (1985) Towards a cumulative Tradition of Research on IT as a Strategic Business Factor. Sloan Working Paper, Centre for Information Systems research, Sloan School of Management, MIT.

223. Tuffe, E. R. (1990) *Envisioning information.* Cheshire, CT: Graphics Press.

224. Van Kirk, D. (1995) 'Poor Asset Management is Breaking the Bank'. INFOWORLD, January 23.

225. Vernadat, F. B. (1996) *Enterprise modelling and integration: principles and applications*. Chapman and Hall. London.

226. Vickers, G. (1984) Changing patterns of communication. In: the Vickers papers, Open Systems Group. Harper and Row, London, pp. 205-216.

227. Vitalari, N. P (1984) The need for longitudinal design in the study of computing environments. In Mumford, E. Et al (eds), op.. cit. In McGolpin, P. (1996) *An examination of the Inter-related factors and issues affecting the degree of success with Information Systems throughout the application lifecycle.* Cranfield University School Of Management Thesis.

228. Vygotsky, L. (1978) *Mind in Society.* Cambridge: Harvard Univ. Press.

229. Wainwright, C. (1996) 'Object-Oriented software development: a case study'. Computer Integrated manufacturing Systems, Vol. 9, No. 4, pp 245-255.

230. Watts, M. and Ebbutt, D. (1987) 'More than a sum of the parts: research methods in group interviewing'. British Educational Research Journal, Vol. 13, pp 25-34.

231. Weaver, P. (1993) *Practical SSADM 4 A complete Tutorial Guide*, Pitman Publishing.

232. Weber, R. P. (1990) *Basic content analysis.* 2nd ed. Quantitative Applications in the Social Sciences Series, Vol. 49, Newbury Park, CA:Sage.

233. Weingand, D. (1993) 'Grounded theory and Qualitative methodology'. IFLA Journal, Vol. 19, No. 1, pp 17-26.

234. Whyte, G. (1994) *Understanding user perceptions of successful information systems*. PhD thesis Cranfield university.

235. Whyte, G. and Bytheway, A. (1995) Factors affecting information systems success. Working Papers. Cranfield School Of Management. 5/95 (SWP/5/95).

236. Wildermuth, B. (1993) 'Post–Positivist research: two examples of methodological pluralism'. The Library Quarterly, Vol. 63, (4), pp 450-468.

237. Winter, H. (1991) ' Liverpool require a New Forward plan'. The Independent, 25 Nov.

238. Wong, E. and Taite, G. (1994) 'A study of user participation in information systems development'. Journal of Information Technology, Vol. 9, pp 51-60.

239. Wragg, E. C. (1980) *Conducting and analysing interviews*. Rediguide 11. University of Nottingham School of Education.

240. Wu, B. (1994) *Manufacturing Systems Design and Analysis, Context and Techniques*. Second edition. Chapman and Hall.

241. Wynekoop, J. and Russo, N. (1997) 'Studying systems development methodologies: an examination of research methods'. Information Systems Journal, Vol. 7, No. 1, January, pp 47-65.

242. Yadav, S. (1981) *Methodology for modelling an origin to determine and develop information systems requirements*. Georgia University thesis.

243. Yaverbaum, G. (1982) 'The technical revolution: Organisations must take positive actions to reduce stress caused by computer's. Journal of systems management, December.

244. Yaverbaum, G. (1988) 'Critical factors in the user environment: An experimental study of users, organisations and tasks'. MIS Quarterly, March.

245. Yin, R. K. (1989) *Case Study Research: Design and Methods.* Sage Publications, London.

246. Young, J. and Kent, H. (1958) 'Abstract formulation of data processing problems'. Journal of industrial engineering, November-December, pp 471-479.

247. Yourdon, E. and Larry, L. (1979) *Constantine, Structured Design.* Prentice-Hall. Inc. Englewwod Cliffs, NJ.

248. Yourdon, E. (1993) *Yourdon Systems Method.* Yourdon Press.

249. Yu, E. (1998) Why Goal-Oriented Requirements Engineering. *Proceedings of the Fourth International Workshop Requirements Eng: Foundation for software Quality* (REFQ '98), Pisa, Italy.

250. Zarley, C. (1989) 'When users write their own applications'. PC week, Vol. 6, (50), December, pp 66-69.

251. Zave, P. (1991) 'An insider's evaluation of PAISLey'. IEEE Transactions on Software Engineering, Vol. 17, No. 3, March.

252. Zwegers, A., Fang, S. and Pels, H. (1997) 'Evaluation of architecture design with CIMOSA'. Computers in industry, Vol. 34, No. 2, pp 187-200.

APPENDIX A

REVIEW OF SYSTEMS DEVELOPMENT METHODS

Review of systems development methods

Structured Systems Analysis and Design Method (SSADM)

Weaver (1993) states that SSADM is the standard Information Systems development method for UK government projects, and has become a *de facto* standard for the UK private sector. It is a structured and rigorous method, with rules and principles that help companies to form detailed sets of information.

The technique is diagram-based and offers the developer a method of representing information in a simpler way. However, it is geared towards the developer rather than the user. The diagrams are used to produce working documents that can represent the current understanding of the system at a given point in time. These diagrams can be up-dated within the development process. The SSADM life cycle fits into the general systems development life cycle as shown in figure A1. The different phases of the SSADM life cycle are then broken down into another set of stages. These are shown in figure A2.

Figure A1: SSADM life cycle (Weaver, 1993)

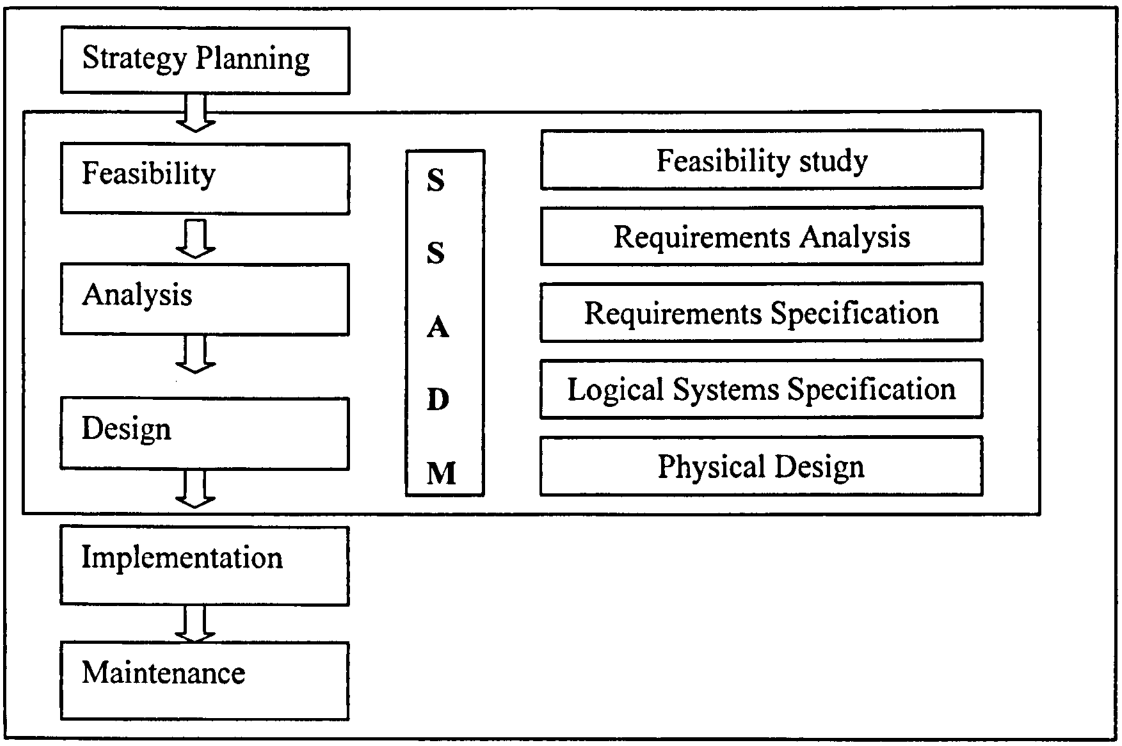
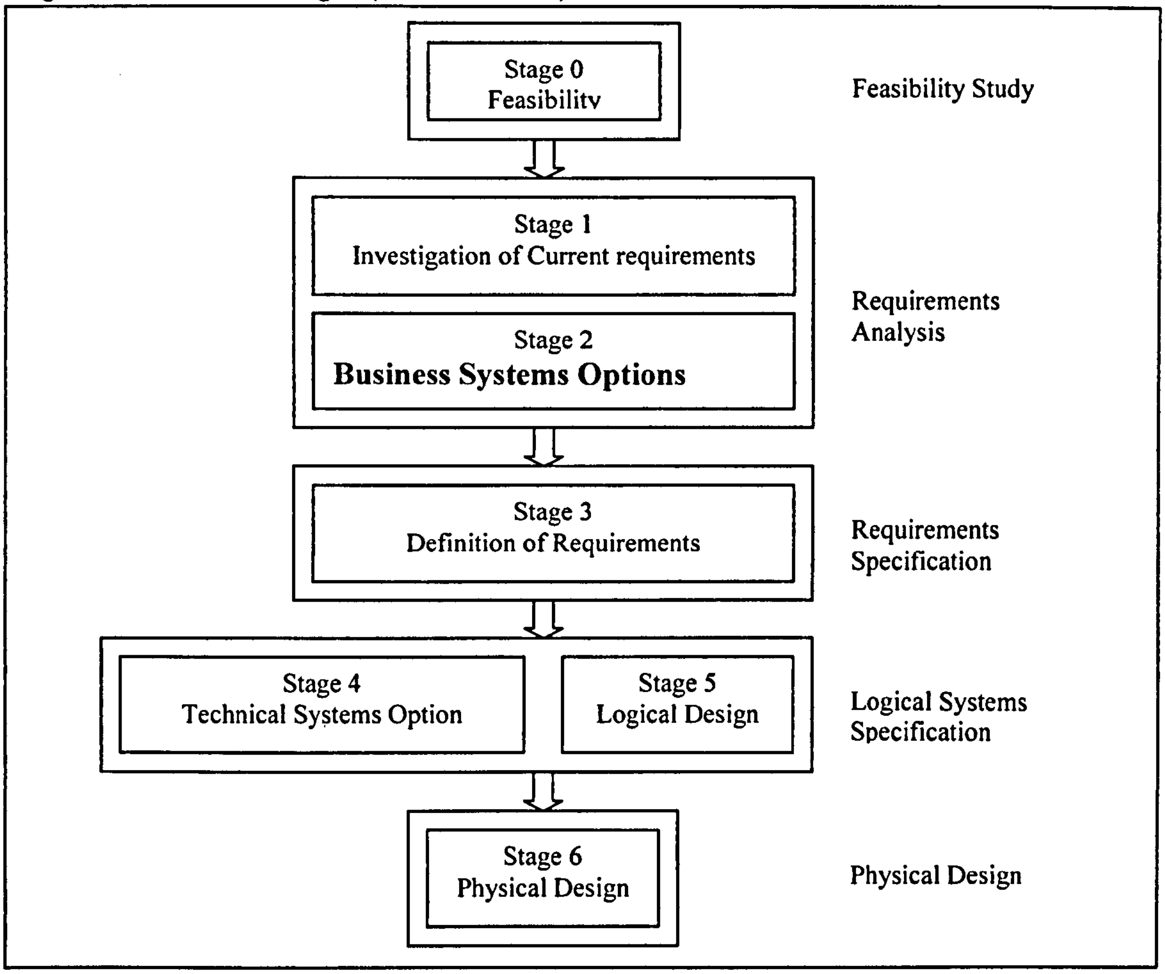


Figure A2: SSADM stages (Weaver, 1993)



Ince (1996) investigated this method and found that there were three views of SSADM, which examines the system from highly independent perspectives. These are as follows: *(1) Functionality or Processing,* which looks at the way that the data is passed around the system and the processes or activities that transform it, i.e. it sets out the functions provided for users by the system. *(2) Data* - An Information System exists only to store and act on organisation's data. This is done by understanding the true nature and structure of the data that is received from the heart of the system. *(3) Effects of time and Real world events on the data held within the system* - This shows that the system is designed specifically to model system behaviour over time.

SSADM uses a set of stages to help guide the developer through the process of systems development. Longworth (1992) states that there are three principle techniques for modelling a system using SSADM, these are: (1) Data Flow Modelling

- Data Flow Models (DFMs) are used in this area. They consist of two parts: Data Flow Diagrams (DFDs) and a set of associated textual descriptions. DFDs illustrate the way that data flows around the system and how it is transformed and stored within the system. (2) Logical Data Modelling - This aims to identify what the system is holding data about. These consist of Logical Data Structures and associated textual descriptions. Entities are related by links on a line from one to another. This line also has a brief statement of what the relationship is between the two entities. (3) Entity-Event Modelling - This will document the events that can affect (i.e. cause a change to or constrain the life of) an entity type. They will also specify allowable sequences and combinations of events. There are a number of concepts and notations that are involved with the ELH (Entity Life History). The ELH shows that there is a choice between one event and another. This is shown by the circle in the top right hand side of the box. These are used on occasions when it is impossible to know the sequence in which events take place.

These examples of the techniques show the amount of structure that the method presents, allowing the developer to represent the data that has already been received from users in a well defined manner.

Dynamic Systems Development Method (DSDM)

Carter (1997) states that DSDM is a non-propriety RAD (Rapid Application Development) method produced by a consortium of which BT is a member, comprising a number of vendors, users and associates. There is an underlying philosophy surrounding DSDM, which Stapleton (1997) has described. This philosophy includes:

(1) Development is a team effort. It must combine the users' knowledge of the business requirements with the technical skills of IT professionals.

(2) High quality demands fitness for purpose as well as technical robustness.

(3) Development can be incremental. Not everything has to be delivered at once, and delivering something earlier is often more valuable than delivering everything later.

(4) The law of diminishing returns implies that resources must be spent developing the features of most value to the business.

The method is based on nine fundamental principles. The first four define the foundation on which DSDM is built and the other five provide the principles that have guided the structure of the method. These are as follows:

(1) Active user involvement is imperative,

(2) DSDM teams must be empowered to make decisions,

(3) The focus is on frequent delivery of products,

(4) Fitness for business purpose is the essential criterion for acceptance of deliverables,

(5) Iterative and incremental development is necessary to converge on an accurate business solution,

(6) All changes during development are revisable,

(7) Requirements are baselined at a high level,

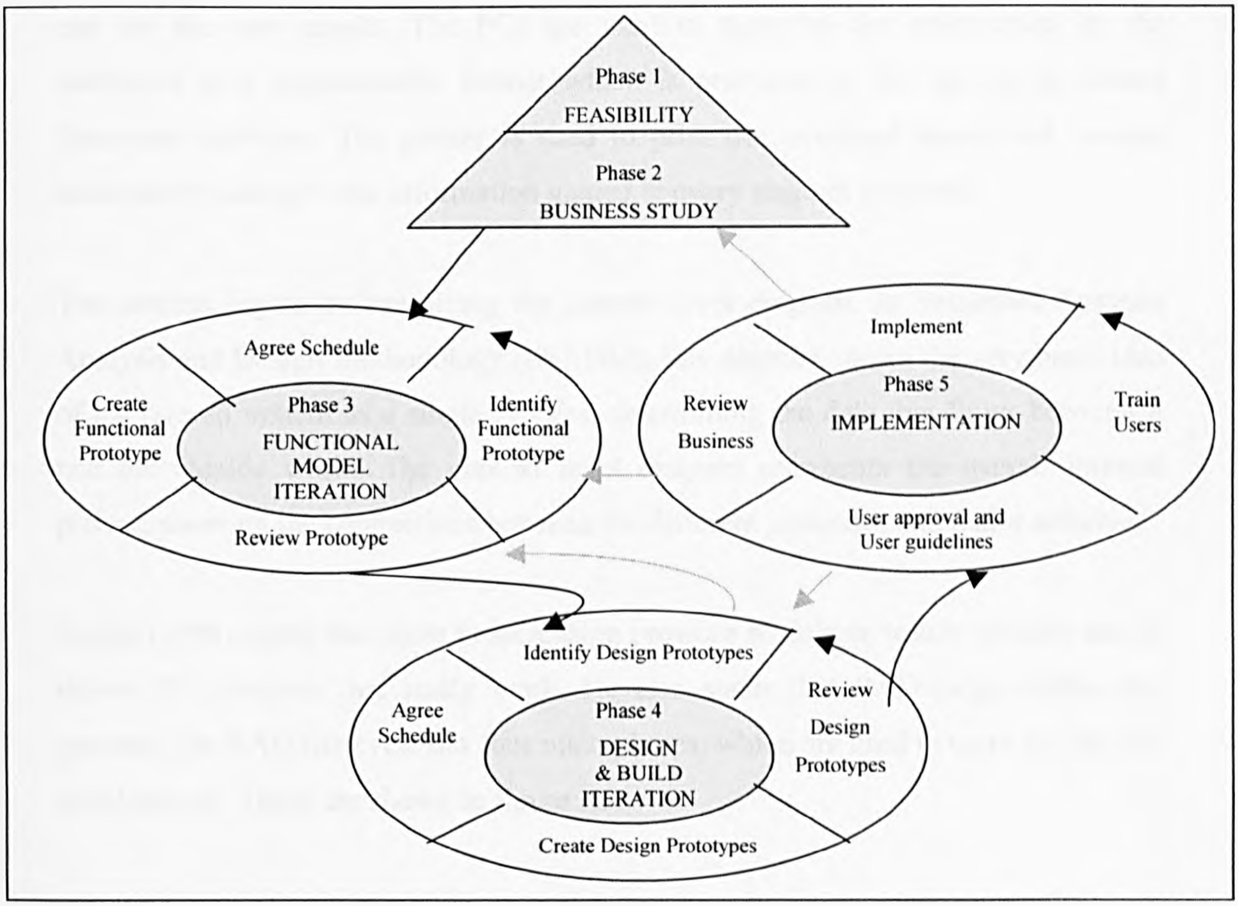
(8) Testing is integrated throughout the life cycle and

(9) A collaborative and co-operative approach between all stakeholders is essential.

The process of development (see figure A3) using the DSDM method is sometimes known as 'the three pizzas and cheese' (Stapleton, 1997). The path that shows the forward approach is shown by the dark arrows and the route backward is shown by the lighter arrows. The whole life cycle is completed in five phases as shown in figure 3. These phases are as follows: (1) Feasibility - There is an assessment of which approach to use for the particular project. The project team may decide that they would like to or should use the waterfall method to cover the whole project or use the DSDM life cycle. It is also important to consider how the problem that has been identified will be solved, (2) Business study - This will be a relatively short exercise. The aim here is to try and get a realistic and adequate understanding of the business and technical constraints to move forward safely. The results of the business study will produce a high-level view of the processes to be automated, (3) Functional model iteration - At this point in the project there will be a focus towards refining the business aspects of the system. The business study is used to build on the high-level functional and information requirements, (4) System design and build iteration - The functional model will be created in this stage. This should be left until this final phase. However, the testing should be carried out throughout the cycle so that mistakes do not go from one stage to the next and (5) Implementation - This is where the system is put in place and users have hands on training.

Stapleton (1997) states that there are a set of rules which are called the MoSCoW rules. These rules were developed by Dai Clegg of Oracle UK, who was one of the early participants of the DSDM consortium. It is an acronym, where the 'O's are not used. The rules are as follows: *M* - *'Must Have'* - for requirements that are fundamental to the system. Without them the system will be unworkable and useless. The must haves' define what DSDM calls the minimum usable subsets. *S* - *'Should Have'*- for important requirements that would probably be classed as mandatory in less time-constrained development, but the system will be useful and usable without them. *C* - *'Could Have'* - for requirements that can more easily be left out of the increment under development. *W* - *'Want to have but will not have this time round'* - for those valuable requirements that can wait until later development takes place.

Figure A3: DSDM process diagram (Stapleton, 1997)



Rapid Application Data Development (RADD), Rapid Application Development (RAD) and Joint Application Design (JAD)

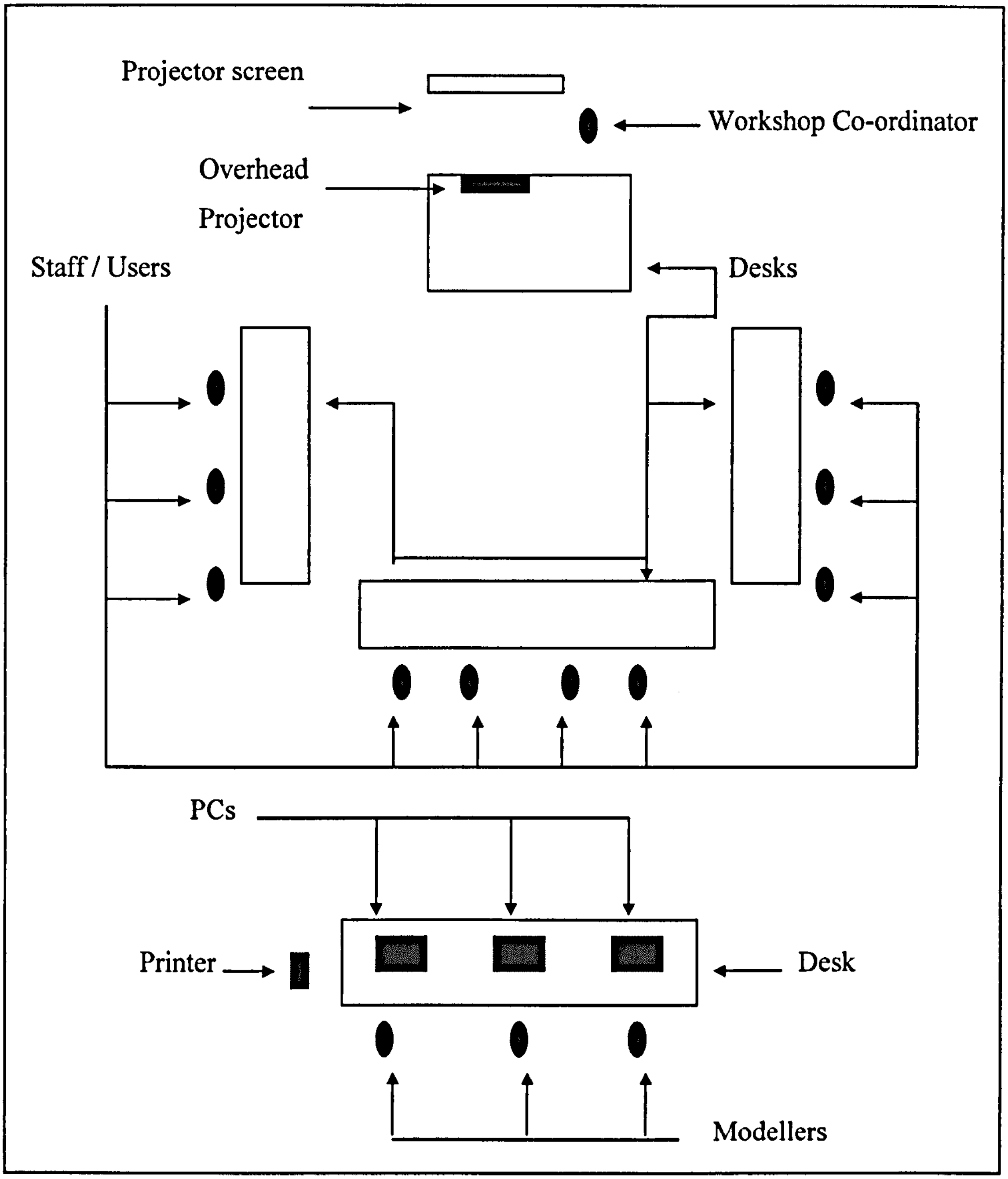
RADD is based on RAD (Rapid Application Development). Figure A4 illustrates how the workshop is laid out with the various items of equipment and the people who are involved. The job of the co-ordinator is to get as much information out of the users as possible, ensuring that it is done in a structured way and the session does not stray away from the specified subject. The staff members will attempt to give out as much information as they can about their specific function within the workplace. The task of the modeller is to graphically represent the information gained in the meeting and also document various definitions that are confirmed on that day. Their requirements also include the preparation of reports and up-to-date diagrams whenever they are required.

The overhead projector and screen are used to visualise the diagrams that represent the system and also for amending the information so that the users and the modellers can see the new results. The PCs are used to represent the information by the modellers in a diagrammatic format which is provided by the use of the Select Enterprise software. The printer is used to print out overhead sheets and various documents relating to the information gained at every stage of progress.

The session begins by examining the context level diagram. In Structured Systems Analysis and Design Methodology (SSADM), this diagram shows the very basic idea of the overall system as a single process, determining the data that flows between it and the outside world. The context level diagram represents the overall internal picture, showing the connections between the different processes and major activities.

Roche (1998) states that there is increasing pressure to deliver results quickly and to deliver IT solutions that really work. He also states that RAD helps within this process. The RAD life cycle has four main phases, which are used to carry out the full development. These are shown in figure A5.

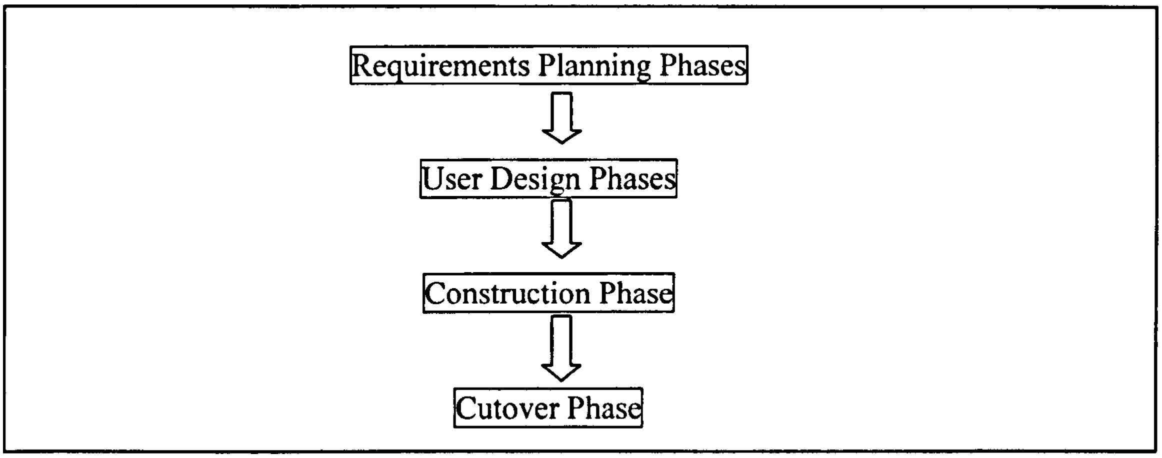
Figure A4: RADD Workshop Layout



RAD is used to push the development process through its paces faster than other methods produced in the past. Martin (1991) states that RAD refers to a development lifecycle designed to give much faster development and higher quality results than the traditional life cycle.

There are advantages of using the RAD techniques. The first is lower cost. Cost reduction may be achieved from shorter development times which may result in the need for less people due to its connection or usage of Computer Aided Software Engineering (CASE) tools, which are used to support the development process. The second is quality. Martin (1991) states that fast development does not mean 'quick and dirty'. It is necessary to build applications of higher quality. However, Martin (1991) argues that much higher quality is needed than is found in many of the applications built by traditional methodologies.

Figure A5: RAD lifecycle phases (Martin, 1991)



RAD consists of a list of the objectives which the methodologies, CASE tools, code generators and Prototyping tools aim to achieve, these include:

A. Meeting Business needs

B. Fit user capabilities

C. Ensure technical integrity in complex analysis and design

D. Create a bug free system

E. Create systems with excellent human factoring

F. Create systems that are easy to maintain

G. Create systems that can evolve continuously, becoming rich in functionality

H. Create systems of great complexity

However, the factors mentioned might assist developers in achieving future systems success. If we start from the first two objectives A and B. This would appear to consider the business and its employees requirements first. Once the present system is documented it would then be beneficial to be fully aware of the business implications before any reengineering takes place toward the development of the proposed system. The user may require many different specifications that do not fit into the business strategy.

C, D, E and F refer to the accuracy of the development and testing process. The main activity is to get the correct information about all of the processes throughout the life cycle ensuring that the testing and continuous feedback is carried out.

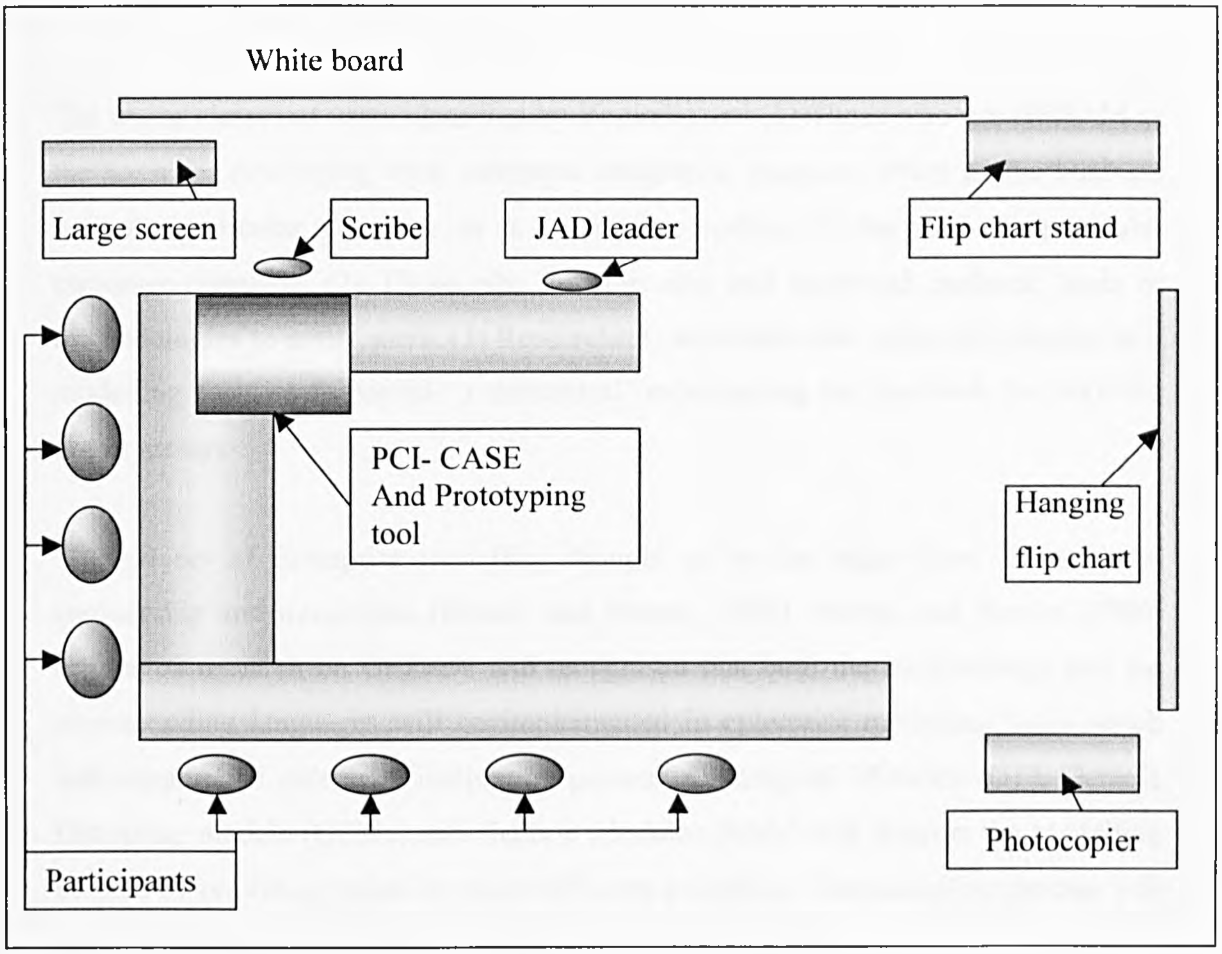
G looks at the future implications of systems development. Change is likely to occur within the organisation. This suggests that once there is an information system in place, it must also change according to the changes occurring within the company. Therefore, the development process should consider that systems might need updating in the future. Therefore, documentation must be clear and concise. H implies that the tools/techniques should complement the development process rather than take it over. This should assist with the users in their ability to feel more comfortable with the systems development exercise.

RAD has four essential ingredients: Tools, People, Methodology and Management. The tools consist of code generators, CASE tools, Prototyping tools and fourth generation languages. The people are those members of the company with the right skills, talents and training to complete a given task. The inclusion of people suggests that they are relevant to the development process. In order to assist the smooth running of the development process, management should ensure that users can be available to be as fully involved as the chosen method requires. This suggests that management should organise the process to ensure that the Red tape and Bureaucracy does not get in the way of smooth development, which should encourage the process to be completed faster. Lastly the Methodology should offer the guidance and structure that is necessary to carry out this process of capturing user requirements. This might assist the process in terms of the planning and repeatability of the project.

Traditionally, specifications were written, the detailed design was drawn, the code was written and finally the code was debugged. With newer technologies such as Integrated - Computer Aided Software Engineering (I-CASE) tools, the detailed design is built on the screen of the tool and the code is generated from it and can be ready to run almost immediately. This enables those end users who are involved in the design to have a closer relationship to the IS team, allowing them to see the design at earlier stages so that important changes can be made before the final implementation. Therefore, the less complicated the diagrammatic format is, the easier it will be for the user to get involved in the processes.

The basic idea of JAD is that the process is carried out using workshops. The workshop usually consists of specifically chosen end users who will go through a process where IS professionals help them to remember what they currently do and what they should do. The IS team will consecutively create the detailed design / specification while the workshop is going on. The layout of the workshop is shown in figure A6. The Joint Application Design (JAD) workshop involves or requires a more detailed knowledge of the system. The data and process models can be created to establish the exact specifications of the system.

Figure A6: An appropriate layout of a JAD workshop



Generic Enterprise Reference Architecture and Methodology (GERAM)

GERAM is a generalised framework for enterprise integration and business process engineering. The methodology identifies the set of components recommended for use in enterprise engineering.

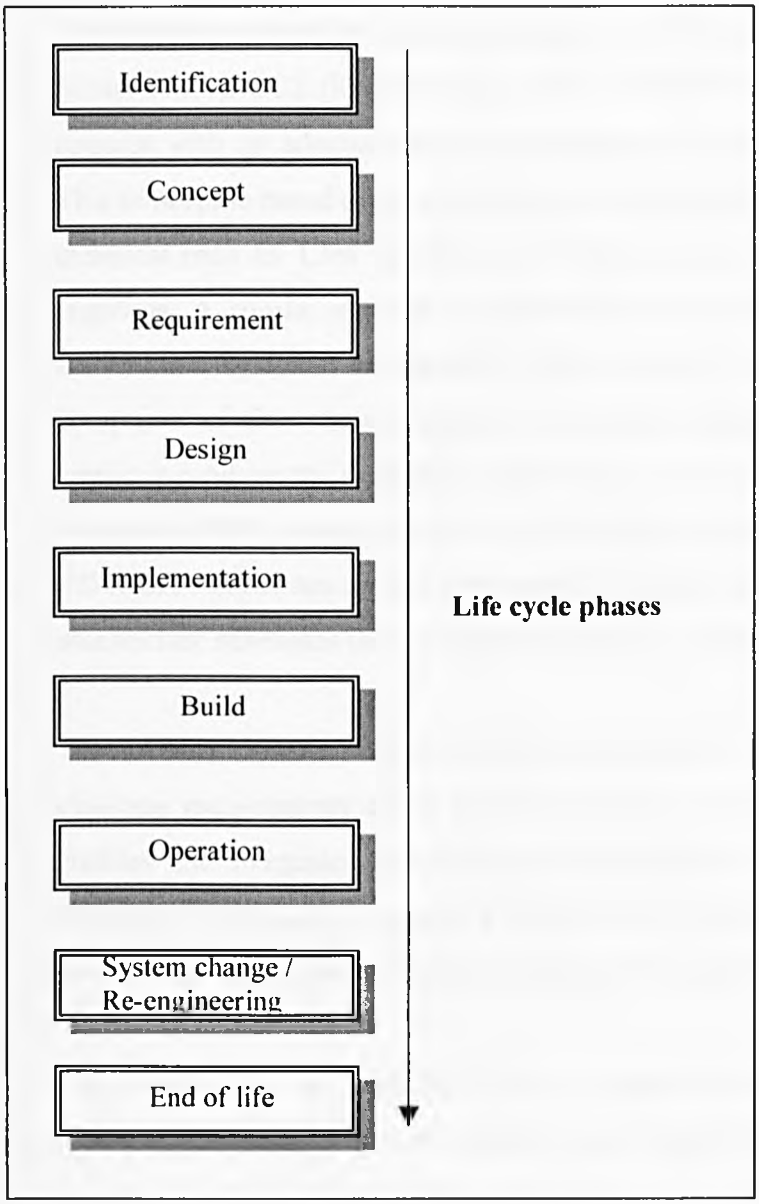
Vernadat (1996) states that GERAM essentially builds on the results from CIMOSA (Open Systems Architecture for CIM), GIM (GRAI-IDEF0-Merise) and PERA (Purdue Enterprise Reference Architecture). He states that the purpose of GERAM is to serve as a reference for the whole community concerned with the area of enterprise integration. It provides a consistent modelling environment, a detailed methodology, promoting good engineering practice for building reusable, tested and standard models, and providing a unifying perspective for products processes, management, enterprise development and strategic management.

The major classes of users identified by Vernadat are: (1) Those who use GERAM as the basis for developing their enterprise integration program, either as the business user of a particular enterprise, or as consultants working for the need of a particular customer company. (2) Those who develop new and improved methods, tools or methodologies to assist users. (3) Researchers, academics and others developing new modelling theories to provide a theoretical underpinning for the work for both the above groups.

The subject of Enterprise modelling thought to be the major item in enterprise engineering and integration (Bernus and Nemes, 1996). Bernus and Nemes (1996) conducted research on GERAM and recognised that both the methodology and the corresponding languages will be implemented in enterprise modelling tools which will support the enterprise integration process. Ontological Theories (OT), Generic Enterprise models (GEMs) and Generic Modules (GMs) will support the modelling process by providing means for more efficient modelling. The modelling process will result in an Enterprise Model (EM) which represents all or part of the enterprise operation. These models will allow simulation of operational alternatives and thereby their evaluation, leading to the optimum structure, contents and behaviour of the enterprise operation. GERAM provides a generic description of all the elements recommended in enterprise engineering and integration.

Figure 7A shows the GERAM life cycle phases of enterprise entities. A total of 9 life cycle phases has been defined. The life cycle concept shows a specific start and finishing point for the developer to work with. However, it is not very different from the life cycle approaches that have been designed in the past.

Figure A7: GERAM life-cycle concept



Computer Integrated Manufacturing – Open Systems Architecture (CIMOSA)

CIMOSA is an Open System Architecture for CIM that was developed by the ESPRIT consortium AMICE (Kosanke et al, 1997). CIMOSA provides a widely accepted CIM concept with an adequate set of architectural constructs to structure CIM Systems. This concept is based on an unambiguous terminology in order to serve as a common technical base for CIM system users, CIM system developers and CIM components suppliers. A similar method to CIMOSA is the ARIS (Architectures for Integrated Information Systems) architecture. Toh et al (1997) state that the ARIS architecture comprises of four views, namely: function, organisation, information and control views. It supports the modelling process from design specification to implementation. Toh et al (1997) also state that the difference between ARIS and CIMOSA is that CIMOSA demarcates the resource model through a major viewpoint, whilst the ARIS architecture subsumes the resource description into information view.

The primary objective of CIMOSA is to provide a framework for analysing the evolving requirements of an enterprise and translating these into a system which enables and integrates the functions that match the requirements. The CIMOSA Reference Architecture contains a limited set of architectural constructs to completely describe the requirement of, and the solutions for, a particular enterprise.

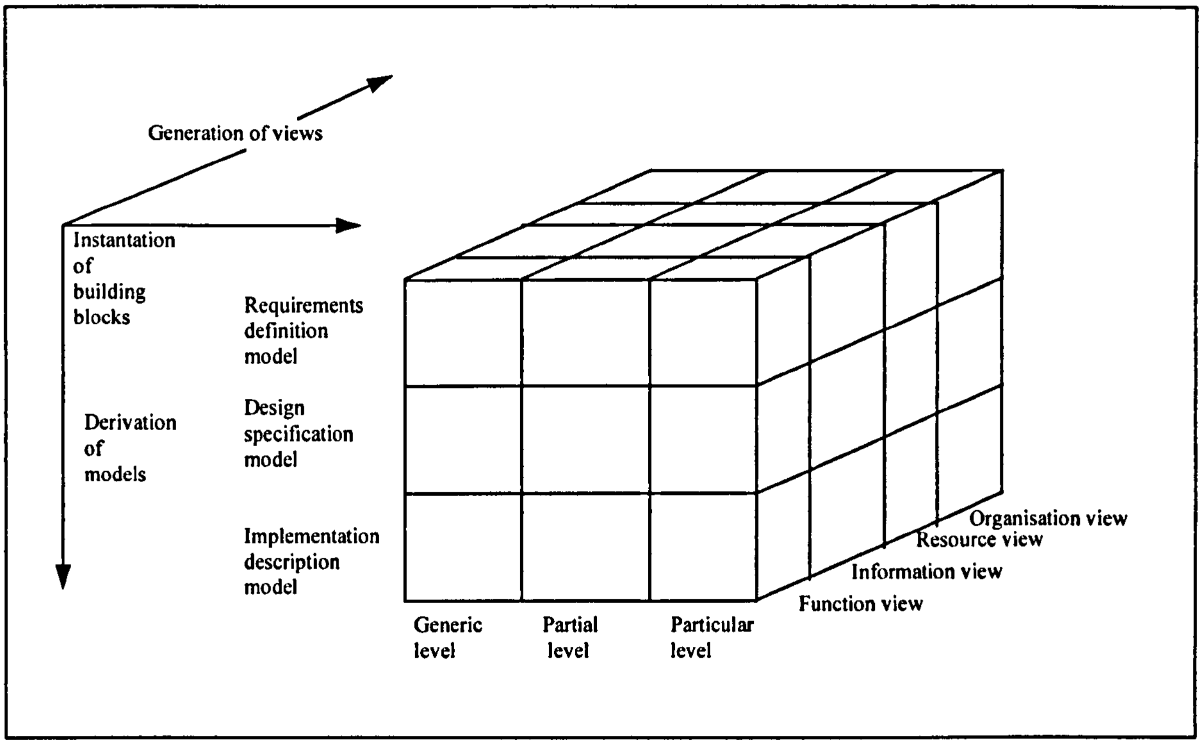
The model will be translated into an Implementation Description Model. This particular model embodies all necessary knowledge about the enterprise in a form that can be directly processed under control of a set of neutral Data Processing services called the Integrating Infrastructure which provides system wide availability of information.

The CIMOSA modelling framework shown in Figure A8 defines three levels of genericity: (1) The Generic level, in which only the generic constructs of CIMOSA may take place, (2) The Partial level, in which models for a class of enterprises and/or industrial domains are potentially embedded and (3) The Particular level, containing the models established for a given domain inside a given enterprise.

The levels of genericity allows one to work with subsets of the model rather than with the complete model, providing especially the business user with reduced complexity for this particular area of concern.

CIMOSA model engineering is relatively complicated and does not really give a defined approach to user input. However, Kosanke (1995) suggests that the method creates a very good template for the development of information systems. The main reason for this is that it divides the project and allows the user to look at the whole environment from different angles, therefore breaking it down into manageable sections.

Figure A8: The CIMOSA modelling framework (Kosanke, 1995)

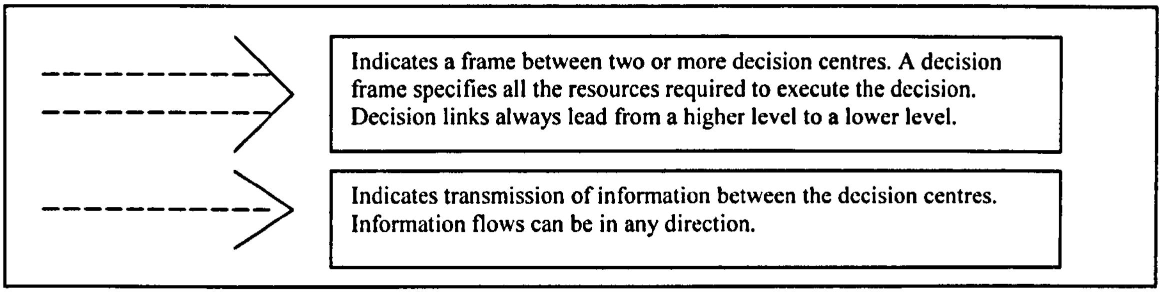


Graphe a Resultants et Activites Interlies (GRAI)

GRAI was developed in the late 1970s by the GRAI laboratory at the University of Bordeaux I in France. The method was developed explicitly for analysing Production Management Systems (PMS). The theory is that decisions start and terminate events within a PMS and these events will determine the performance and operating characteristics of the system (McCarthy et al, 1994). Wu (1994) states that the GRAI method is an approach which has been specially developed for the design of managerial systems and consists of two main tools: the GRAI Grid and the GRAI net.

The GRAI grid is used to show the links between the main decision centres. The grid uses columns, which represents the various elements, for example, to buy or to make. The levels have a corresponding horizon and review period. The relationships and structure of the decision centres are indicated by two symbols as shown in figure A9.

Figure A9: Symbols describing the relationships and structure of decision centres (Doumeingts, 1985)

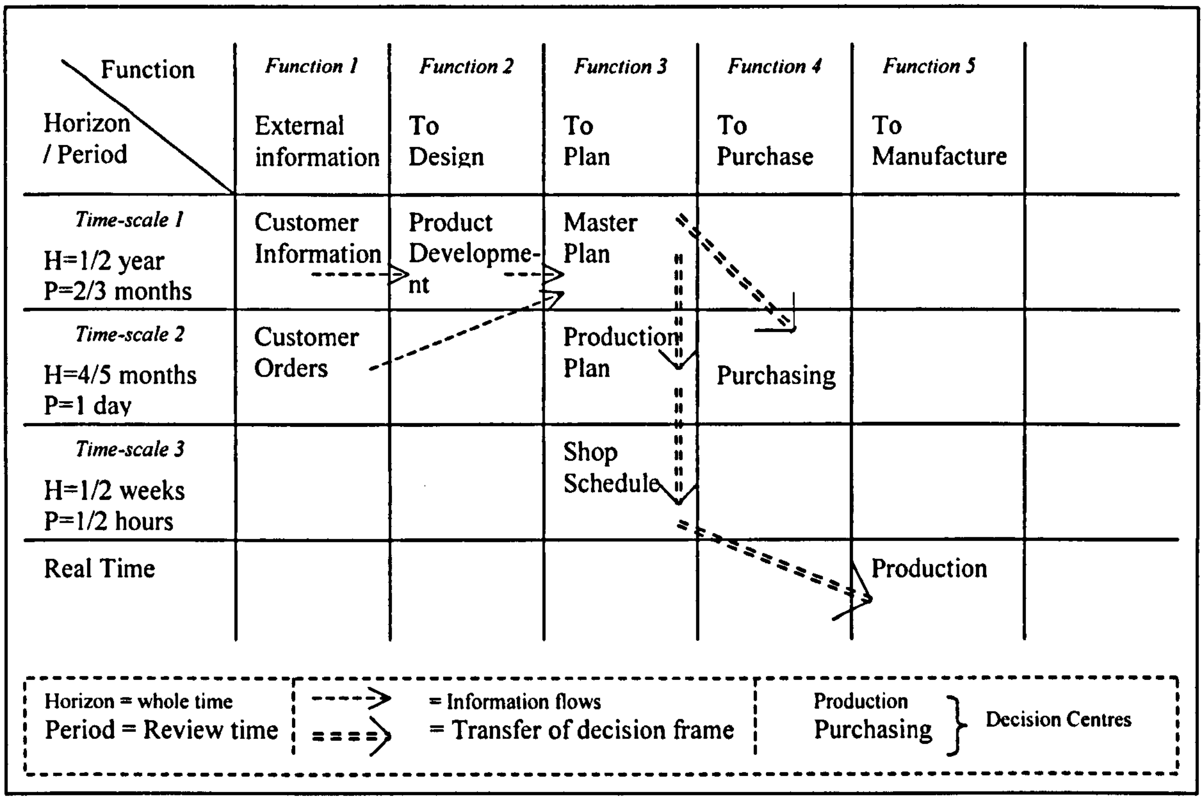


Wu (1994) also shows how the grid is structured. This is shown in figure A10. It can be observed that a structured set of functions are represented in the columns of the grid, while the desired timescales are represented in the rows. The squares are considered as being a potential Decision Centre (DC) in the related function and time-scale. These will be examined in the context of the desired system, to establish whether there is a set of decisions to be made in this response period (Wu, 1994).

GRAI nets examine the micro structure of the system. The net is drawn for each decision centre. This is done by analysing the components within the decision centres. A GRAI net is an illustrative portrait of a decision, activities and their interrelationships (McCarthy et al, 1994) This is shown in figure A11.

GRAI uses a top-down and bottom-up approach to define the decision centres and their information. It represents the basic concepts of a manufacturing system (Chen et al, 1997). A manufacturing system is decomposed into three sub-systems: physical system, whose role is to transform raw material into products; decision system which manages and controls the physical system; and information system, which provides the decision system with information needed for decision-making.

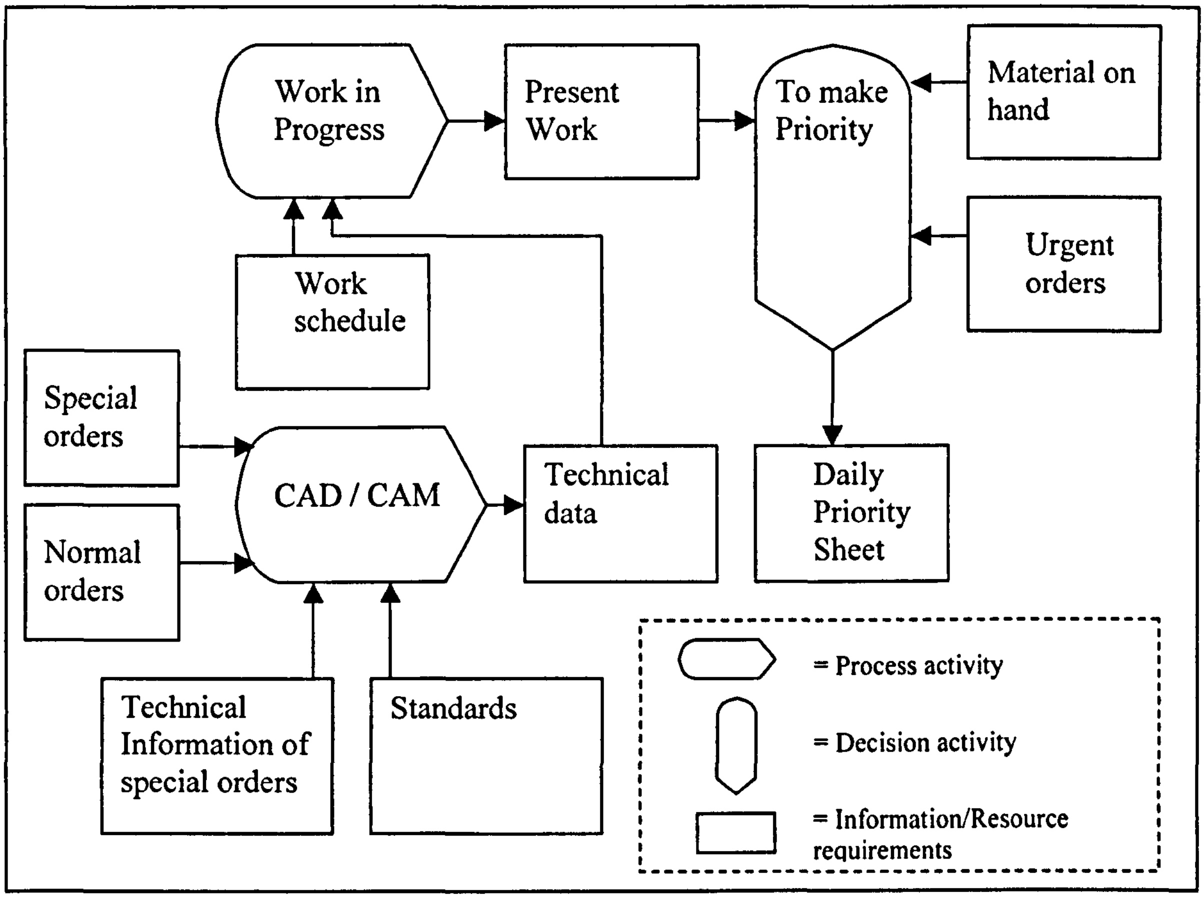
Figure A10: Structure of a GRAI grid (Wu, 1994)



The Decision Centre (DC) is an important part of the GRAI method. It is a sub-system of the decision system within the manufacturing system. Doumeingts et al (1993) state that it can be decomposed at any level of the structure using the same criteria of decomposition: a physical element, a decision element and an information element

There are a set of stages that Komi (1995) has identified as being the stages of GRAI modelling; these are as follows: (1) Identify the management team involved in decision making, (2) Build a synthesis group of managers involved in decision making, (3) A top-down analysis is performed by holding a meeting with the synthesis group in order to construct a GRAI grid, (4) A bottom-up analysis is performed to construct a GRAI net for each decision centre. This is done by interviewing each member of the synthesis group separately and (5) A model of the company is used to identify inconsistencies in information and decision making systems.

Figure A11: GRAI net- Release Daily Priority Sheet (McCarthy et al, 1994)



ICAM Definition (IDEF0)

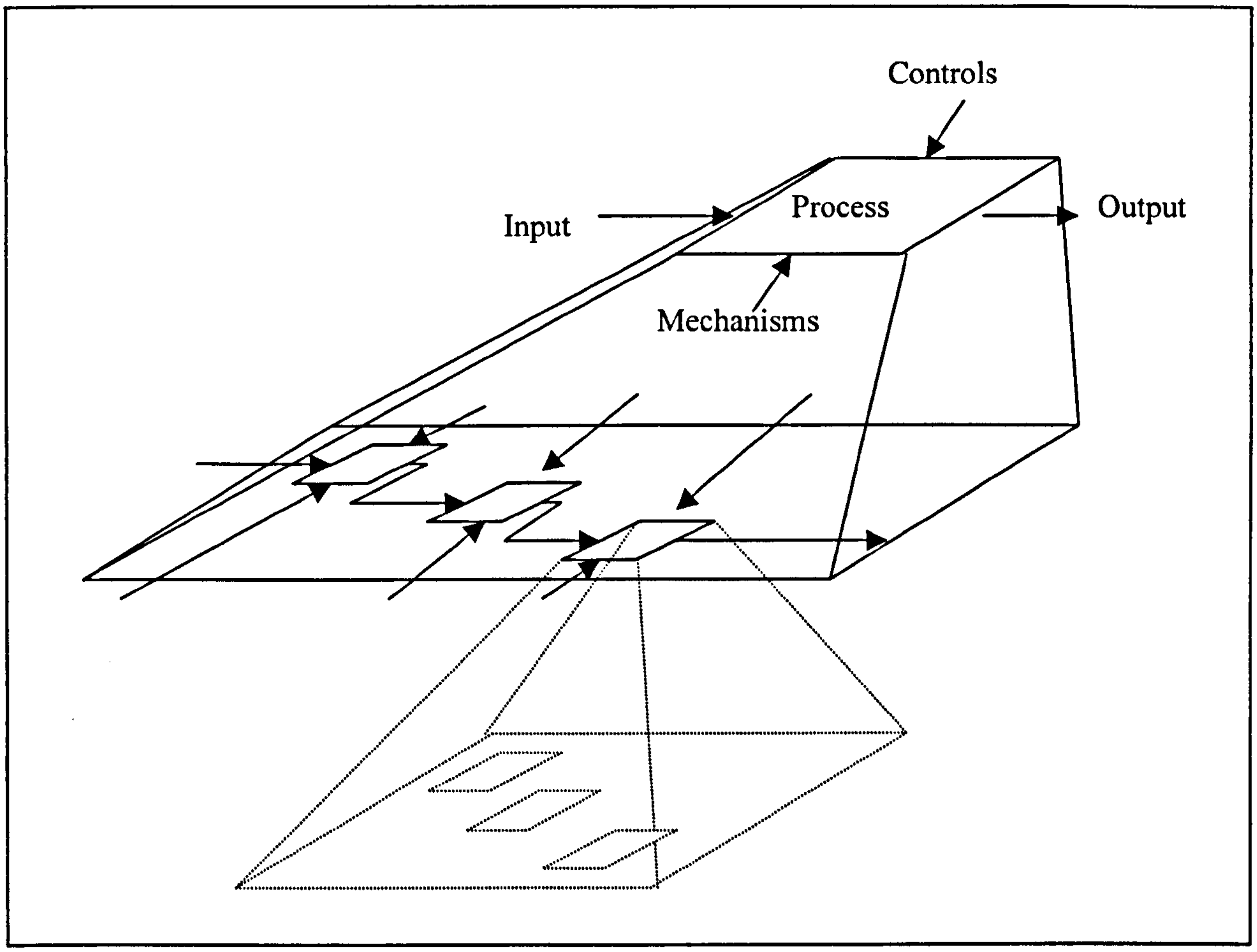
Busby et al (1993) state that a critical aspect of IDEF0 model authoring is its subjective basis that the analyst naturally tends to cast upon the model, based upon prior experience and the way with which he or she is accustomed to interpreting and looking at in reality. Cantamessa et al (1998) states that this phenomenon will be enhanced in the case of low involvement by the organisation members. It is shown that there is a great deal of importance placed on people and their involvement. This suggests that IDEF0 tends to be geared towards involving users in its approach to development.

Cantamessa et al (1998) also state that IDEF0 modelling identifies inconsistencies in information flow that are commonly found in manufacturing systems (i.e. activities that receive more or less information than required, or that produce information which goes unused). However, Komi (1995) states that the structure of IDEF0 is very similar to SSADM. They both use a top-down approach, which describes the system investigated as one process presented by a single box. The process will then be divided into sub-processes and these sub-processes are further divided until the required level of information about the system has been achieved. This is shown in figure A12.

The basic element of the IDEF0 model is called a function block. This is shown in figure A13. Wu (1994) states that the inputs to a function entering the function block from the left are usually (but not necessarily) 'consumed' by the function to produce outputs. Raw materials are typical examples of these. The mechanism, represented by the arrow entering the function block from below, indicates the resources required to carry out the transformation process – such as tools, equipment and operators. All resources shown must be used as a means to achieve the function. They only become part of the output if they are passed on by the function to support other functions. Finally, the controls, which enter from the top of the block, only influence the transformation process and they will not be consumed or processed themselves. It is important to remember that all inputs, controls and resources must be used by a function.

The description given by Catamessa et al (1998) of what are considered as being the main elements of IDEF0 are as follows: (1) Activities – (IDEF0 'boxes') elementary units which makes up business processes. Each activity is characterised by its inputs and outputs, (2) Entities and Information – (IDEF0 inputs, outputs and controls) processes are made up of a flow of entities (e.g. parts, raw materials, fixtures etc.) and information. (3) Resources – (IDEF0 Mechanisms) activities are executed by the intervention of resources, be they human or machines, which are described in the intermediate model. Each resource is directly related to entity and information flow among the activities they are involved. (4) Departments – each resource is associated with a company department.

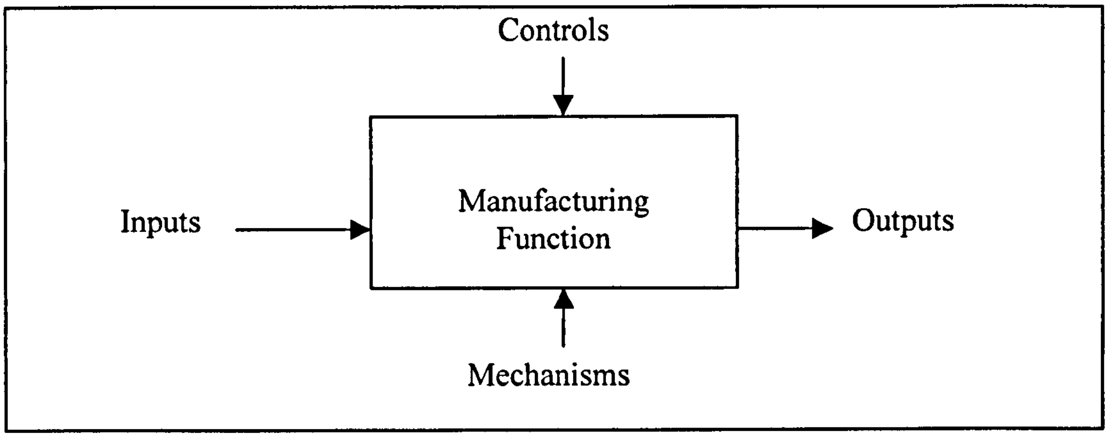
Figure A12: Hierarchical structure of IDEF0 model (Komi, 1995)



The IDEF0 modelling approach can be shown in a systematic format, which involves a set of phases to be completed from the beginning to the final goal. These are as follows: (1) Phase 1: Building the 'As-is' model - The current situation is modelled in this phase. (2) Phase 2: 'As-is' model analysis and 'Should-be' model building - The current deficiencies and limitations will be seen from the development of the 'As-is' model. This should help to produce the 'should-be' model according to the aims of the proposed system. (3) Phase 3: Definition and Implementation of the 'migration plan'-

The 'As-is' and 'Should-be' models are compared at this stage. Modifications may still occur. They normally come in the form of organisational, procedural and related information systems. During this stage project management techniques may be of great help. Also to help with the 'migration plan'. (4) Phase 4: Auditing of the process - the purpose of this phase is to verify each of the previous stages. The aim will be to initiate a continuous improvement process which may continue long after the present interaction has been completed.

Figure A13: Basic building block of an IDEF0 model: The Function (Wu, 1994)



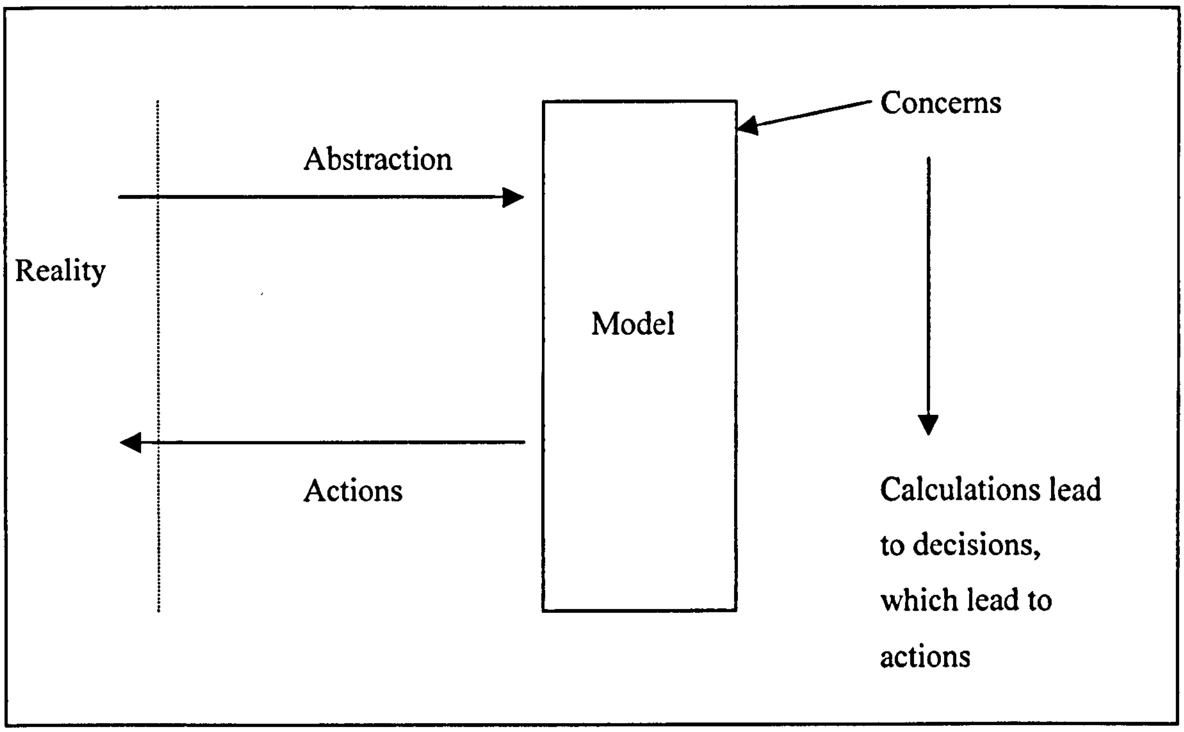
Yourdon Systems Method (YSM)

Avison et al (1995) state that functional decomposition or top-down design, in which a problem is successively decomposed into manageable units, was the basis of the YSM approach. However, the latest version of YSM (3.0) is neither pure top-down nor bottom-up, but is described as middle out. The analyst can design by drawing a top-level context diagram that indicates the system boundaries. Then, following interviews with users, a list of events in the environment that the system must represent is constructed.

YSM is a technique that makes effective use of models. These models are used to represent the main features of what is being examined. These will eventually help decisions to be made. Figure A14 demonstrates this decision making process in relation to the use of the model.

The term model is used in this case to specify a specific structure that is complete and changeable. Each model will have a well defined structure. This helps the model manipulator to check it for completeness and consistency.

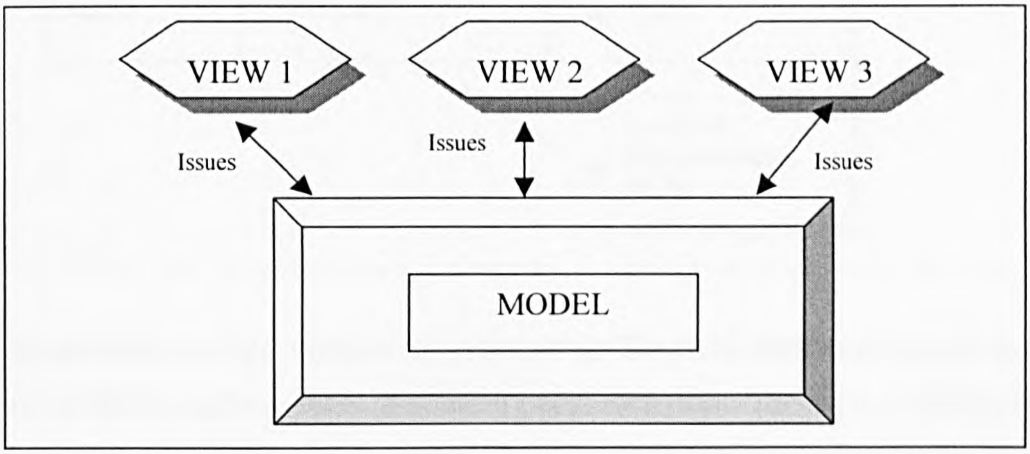
Figure A14: Models and decisions (Yourdon, 1993)



The model also uses a term called the VIEW CONCEPT (see figure A15). The view represents some form of diagram, table or specification that is used to highlight any area of concern. Further use of the view concept helps with the discussion of various sets of issues. Eventually, once the views are analysed, they will then be able to cover all aspects of the documented system requirements. Figure A15 shows how the views are related / connected to the models. The view is used to extract vital information from the model and then expand on that information to make it as user friendly as possible.

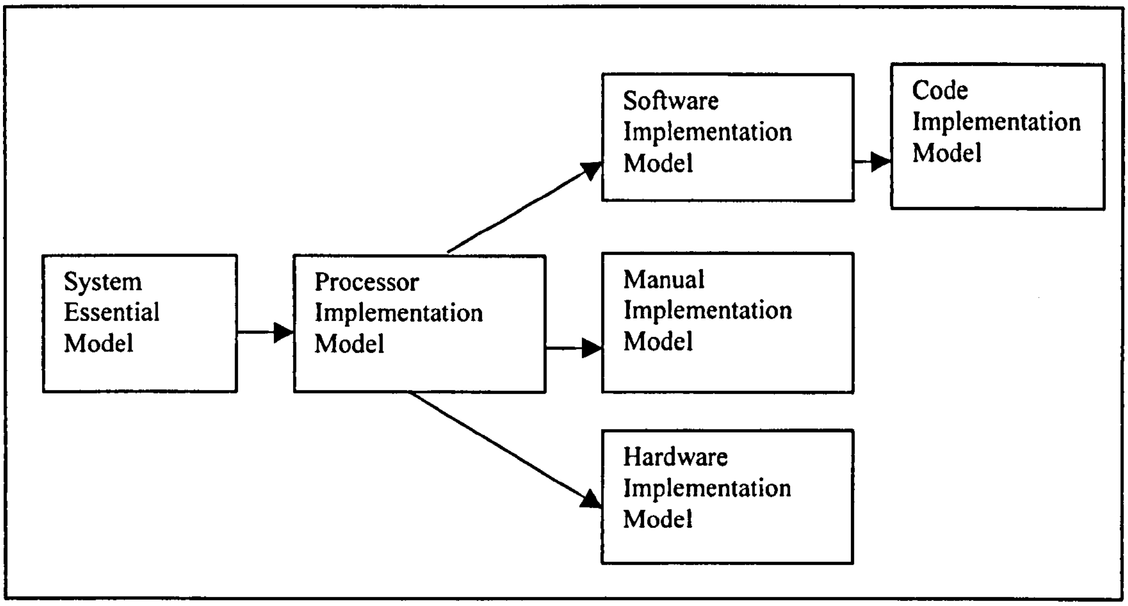
The life cycle of this method involves a sequence of decisions. This life cycle is shown in figure A16. The first model, which is called the System Essential Model, is used to capture policy requirements that are extracted by the user. This should not contain technical implementation details. The other models concentrate on how the organisation will be affected by the implementation of new technology. The Systems Essential model and the Systems Processor Implementation model have a closer link to the subject of capturing user requirements. The other models lean more towards the overall organisational implementation.

Figure A15: Context diagram (shows flows of information in and out of the system) (Adapted from Yourdon, 1993)



Yourdon (1993) states that the most important factor to consider during the systems essential model is to find out what the user will need. The method encourages the user to find the most cost-effective way of carrying out the process. Therefore, a major consideration is the lack of involvement or connection with technology. This method produces the system and allows it to run in a 'perfect conceptual processor' with zero cost and instruction time, and infinite memory and mean time between failures.

Figure A16: Lifecycle of the YSM (Yourdon, 1993)



The processor and the systems requirements are linked in order to determine how they will perform together or how they can support each other. The main consideration will be to address the ability of the processor to meet the systems requirements taking into account the capabilities of the processor and other essential requirements that have been allocated to them. This means that the processor capabilities and capacities are examined to ensure that they can meet the performance requirements.

Object Oriented Analysis (OOA)

OOA is a process to abstract essential entities in an application domain, and to explain how they interact with one another without regard for eventual implementation (Booch, 1991). Wainwright et al (1996) state that the main concerns of OOA are the identification of significant entities (perspective of the classes and objects in the real world problem), and an understanding of the interaction among them.

Coad and Yourdon (1991) state that OOA consists of five major activities that should be performed, they should not be seen as stages or sequential steps. They point out that many analysts prefer to iterate the various activities in a variety of sequences. Avisson et al (1995) state that an object in OOA is an abstraction from the problem domain, about which we wish to keep information (attributes of the object) and with which we can interact (the services). A class is a description of one or more objects with a common set of information and interactions. This suggests that data received should be accompanied by a description to justify its validity within the proposed system.

Hoche et al (1996) state that OOA adds the principle or procedural abstraction used in functional decomposition to other principles aimed at managing the complexity of systems. Object Orientation has many significant advantages over other analysis and programming concepts, in that it allows better complexity handling, modularity, data abstraction and reusability and extendibility of components and software. Coad and Yourdon (1991) state that OOA layers model the problem domain and the system's responsibilities, and consequently represent the core of the object-oriented paradigm.

The use of OOA is quite distinct. The boundaries between analysis and design are not very clear. In analysis the aim is to model the world by discovering the classes and objects that form the vocabulary of the problem domain. In the design stage the abstractions and mechanisms that provide the behavior that this model requires are developed. (Booch, 1994)

Structured Brainstorming (SB)

Structured Brainstorming (SB) is based on the eliciting of user requirements for products or services. SB collects information for the design of new and different solutions by encouraging thinking beyond the boundaries of current problem solutions and traditional methods of accomplishing objectives (Byrne and Barlow, 1993). There are 4 major exercises, which are carried out within the method. These are methods to: (1) Overcome resistance to change or enhance creativity. (2) Evaluate todays products. (3) Develop ideas for future products. (4) Assign priorities to those ideas.

There are also a further set of considerations, which include: (1) Avoid being victims of habit, just because something "is" does not mean it "has to be". (2) Push yourself. Try to generate a broad scope of ideas and investigate the depth of one idea. (3) Avoid judgement. Do not evaluate ideas until all of the ideas have been collected, since early evaluation may prohibit productivity. (4) Avoid functional fixation. Do not limit your thinking by defining something based on its existing function. (5) Relax and have fun. Generating creative ideas occur best in a relaxed, permissive environment.

The main points that give this technique its validity in relation to capturing user requirements are firstly its approach to open-mindedness. Secondly the technique focuses on improving the current system rather than repeating the current system specifications.

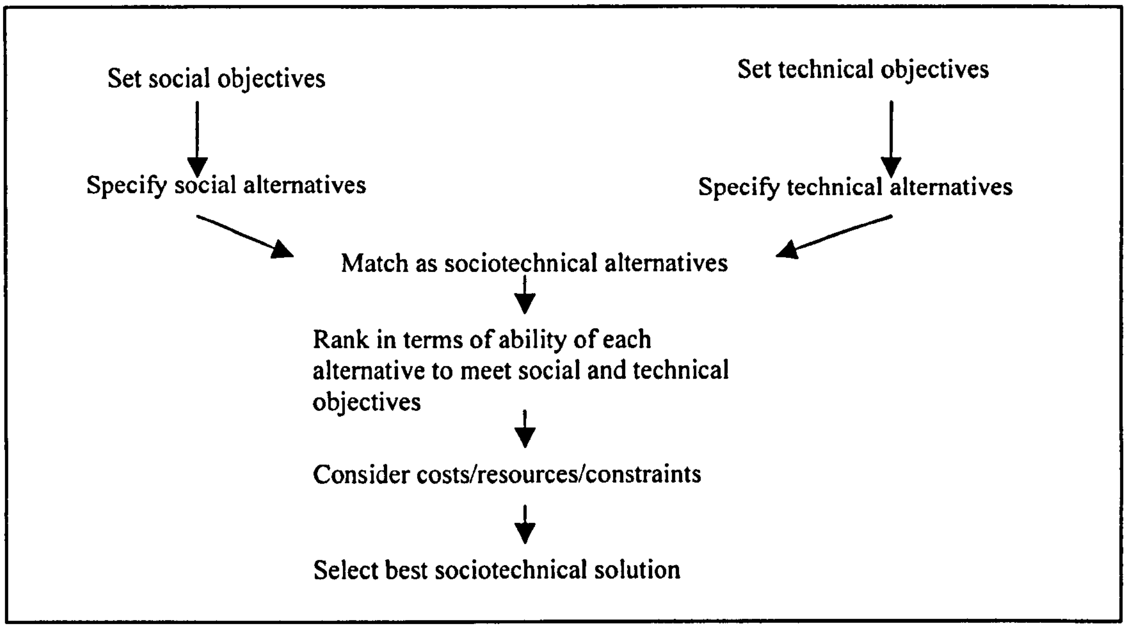
Effective Technical And Human Implementation of Computer-Based Systems (ETHICS) Method

Mumford (1994) states that user participation in systems design is evaluated as a means for ensuring that all with an interest in a system are in a position to discuss and communicate. This suggests that the people who are going to use the system should be participants in the design process.

Wong (1994) states that user participation in information systems development is considered to be an important factor influencing implementation success or failure. Wong (1994) reviewed the ETHICS. Wong (1994) argues that the approach views the interaction between technology and people as important for producing systems which are both technically efficient and lead to high job satisfaction. This suggests that it emphasises users as well as expert participation. The sociotechnical design process is shown in figure A17. This examines the objectives that are set and how these objectives are transformed into a solution.

The ETHICS method considers the social side of the development process and helps to structure the objectives and the path towards a successful development and end result. The feedback process seems to be lacking due to the method concentrating on how everything should be done, and looking at what should be considered and not enough on how you do it. However, the method does emphasise the use of people to a large extent.

Figure A17: Sociotechnical systems design (Mumford and Weir, 1979)



APPENDIX B

USER QUESTIONNAIRE

(SURVEY OF USERS AND DEVELOPERS)

QUESTIONS FOR USERS OF SYSTEMS. TOGETHER WITH THE REASONING BEHIND THE QUESTIONS

AIM

This area of the study aims to create a good idea of what users within the company actually think about the development methods currently in place. It will also establish what they would be happy with in relation to the level of interaction that they would expect or the type of interaction that would help them to use the system more efficiently.

Section 1

1. Are you happy with the systems currently in use within this organisation?

This is to obtain a basic understanding of the users overall feelings towards the systems in the organisation.

2. If No, what is wrong with the systems?

The direct approach to finding out the particular good or bad points about the system is used here. This will ensure that the participant has a clear decision made to establish whether they like the system or not, and what is actually good or bad about it.

3. If Yes, what makes it so good?

Same as 2

4. Is there adequate functionality in the system for your job within the company?

If the previous question did not mention functionality, this question aims to achieve some information regarding this subject. Functionality is one of the main aspects of the study. Users must be clear about what the system should do and how it should do it.

5. Do you remember how the systems were developed?

At this point the actual development process is brought to the forefront of the discussion. The participant must be aware of the development process. This will be established at this point or the discussion must go to the next section which relates to what the user would like in relation to levels of participation and satisfaction.

6. If Yes, how did the process take place?

The actual development process must be explained here. This is where the user must recollect the development process. This will help to establish a good idea of how the system was developed.

7. If No, go to Section 2

If the user does not know about the process. This discussion will go to section 2.

8. How much interaction was there between yourself and the development team, for example: How often did you meet?
How long did you meet for?

The level of interaction is important to establish. This question is quite open and will be useful for achieving the personal opinions of what the user actually thinks without giving them too many options.

9. What percentage of the users actually took part in the development process?

0---100%

This question aims to find out if the system actually considered all of the users in the development process. If they did then the study must be able to use this information to establish what percentage of users actually need to get involved to produce successful systems.

10. How do you think that the methods that were used could be improved?

Here the user should express their opinions on how the system should be developed. These suggestions will be included in the method once they are feasible.

11. Would you like more interaction between yourself and the development team, and why?

This aims to achieve a rough idea of why users actually like to have more interaction or less interaction.

12. Would you like less attention from the development team, and why?

Same as 11

13. Would the fact that you were involved increase your willingness to use the system?

This question is used to find out how user involvement relates to user motivation. These aspects are necessary for the study. Motivation is a very important factor, as the users need to use the system on a regular basis. Therefore, it would be better if they are happy with the system and that they understand what they are doing.

Section 2

1. Do you think that the system currently in use is needed, for example:

Were you consulted?

Does the system do what it is supposed to do?

Does the system help to achieve the result that it was intended to achieve?

This highlights a very important point. In this instance the user can attempt to answer three questions. The first would be, were they consulted when the development was taking place. The second would be, do they feel that the system does what it was supposed to do and the third would be, does the system help to achieve the job that it is intended for.

2. What does satisfaction mean to you?

This will allow the study to obtain some first hand knowledge of how users define satisfaction. This can be used to measure the success of the system from the users point of view.

3. What levels of satisfaction are you experiencing?

0--100%

This question will be used to form some statistics on the levels of satisfaction achieved by a system actually being put in place. This can be used for analysis purposes.

4. What level of dissatisfaction are you experiencing?

0--100%

Same as 3

5. By what percentage do you think that user involvement should be increased?

0--100%

Here the aim is to establish how much involvement the user would feel happy with. The greater the increase or decrease the greater the need for change.

6. By what percentage do you think that user involvement should be decreased?

0--100%

Same as 5

7. What factors would help you to be more enthusiastic about using the system?

This question is really used to establish what types of attributes actually lead to the satisfaction for the users.

8. Would you like the system to be developed in a:

Group
Individually
One representative
Group then a Representative
Other

This question aims to establish a consensus opinion on the personal dynamics of achieving a good system. This will form the methods in which the new method can adopt to carry out the process of capturing user requirements.

APPENDIX C

DEVELOPER QUESTIONNAIRE

(SURVEY OF USERS AND DEVELOPERS)

QUESTIONS FOR DEVELOPERS WHO INTERACT WITH USERS TO OBTAIN USER REQUIREMENTS. TOGETHER WITH THE REASONING BEHIND THE QUESTIONS

AIM

The aim of this section of interviews is to achieve a better understanding of the methods that companies use to capture user requirements.

The following section demonstrates a set of questions which are used in the meeting with the developers / users requirements analysts. These questions aim to bring out some valid points regarding methods used for capturing user requirements.

1. How are user requirements identified?

This question should open up the question about user requirements. Allowing the interviewee to begin thinking about the subject in more detail.

2. What methods are used to acquire user requirements?

This question aims to obtain some detail about the actual method used. For example the structure and content of the method, especially if a formal structure is used.

3. What is the time limit on the analysis and specification process?

This question can be used for testing purposes. For example, if a two week period is used to interview 10 employees and form an analysis and specification. The new method can be compared to this in a controlled experiment.

4. How would you improve this method of development?

The aim here is to get the developer to create ideas that would help them further in the user requirements capturing process. These ideas (once feasible) will be added to the method being developed. Suggestions such as; how would you make it faster, more reliable, cheaper etc. will be given.

5. Who are the users of the systems that are produced by the systems department?

Here the users are identified to establish the level of ease or complexity required in the method.

6. What are the main problems in the requirements capturing process?

Here suggestions can be sought to help departments to have a better understanding of what aspects of the capturing process will help the developer to obtain better quality information.

7. How is the analysis process carried out?

Here the developer can express what happens with the data once it has been collected. This will help in the development of the new method as various points may need to be addressed while attempting to create a practical solution. It will also be possible to establish how much interaction the user has in the analysis stage.

8. Do you always get the right information after the first interview?

This aims to establish the frequency that the developer gets quality information from the end-user. Then it will be possible to find out why this occurs and how it can be improved.

9. What are the main problem areas of the systems that are currently in place?

This will highlight the types of problems that companies are having today with present systems. This will be a result of past methods of developing systems.

10. How do you know that the information obtained from the users is correct?

Here the aim is to find out how the company tests the system development success.

11. How long are the systems expected to last?

This question aims to find out the life expectancy of the systems and the reasons why they last for theoretical reasons. The main aim here is to find out what methods have been put in place to ensure that the systems last a reasonable length of time and how they deal with changes to the system.

12. How much technical knowledge of the development process should users have?

This will show the developers view of how to interact with users. Whether to be technical or non-technical.

13. What methods are currently used to determine the final success that the system will produce before the actual implementation?

This will highlight the testing methods actually adopted by the company. These can also be used for the research testing process.

14. How is the success of the system measured?

This aims to find out how the company can tell that the systems that they have produced is a success. This can be used to test the success of the system produced using the new method.

APPENDIX D

URCAM DEVELOPMENT FORMS

URCAM
DEVELOPMENT FORMS

SESSION PREPARATION

Note: Positions = jobs descriptions / roles / job functions

1. The developer must have an estimated figure regarding how many, a) Positions, b) Departments, c) Systems and Manual Stores that will be used in the process.

This will enable the developer to have a rough idea of how the forms will be laid out and how many forms are required.

(It may be sensible for the developer to have at least 50% more forms prepared for usage.)

This will also enable the developer to give the, a) Positions, b) Departments, c) Systems and d) Manual Stores codes which can be used throughout the process.

The aim of this process is to allow the developer to be prepared for the sessions

AS - IS

Example

Questions

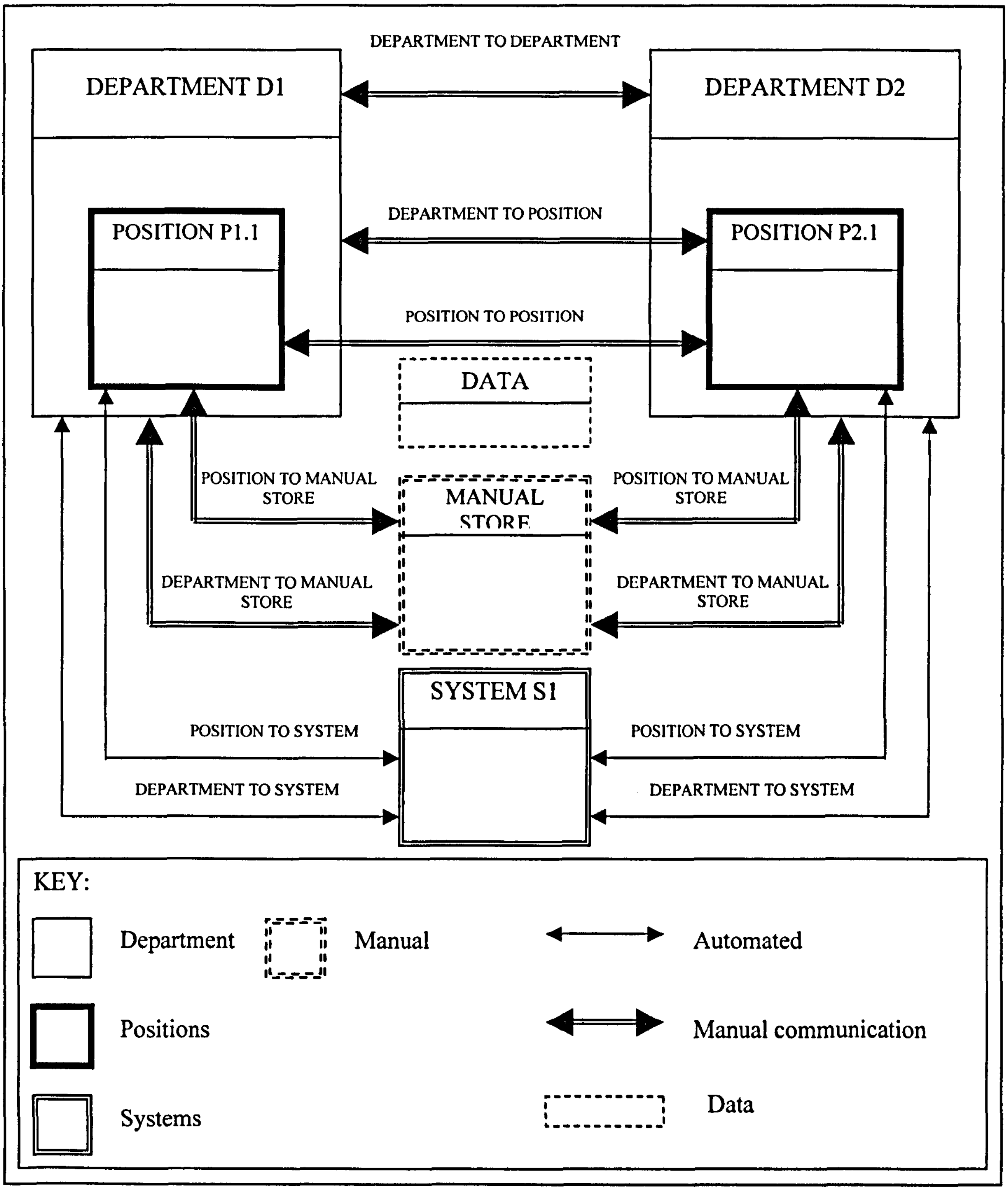
1. Name of your a) POSITION / b) DEPARTMENT / c) SYSTEM / d) MANUAL STORE
2. Name of the other a) POSITIONS / b) DEPARTMENTS / c) SYSTEMS / d) MANUAL STORES that you communicate with.
3. What data do you send to the other a) POSITIONS / b) DEPARTMENTS / c) SYSTEMS / d) MANUAL STORES.
4. What data do you receive from the other a) POSITIONS / b) DEPARTMENTS / c) SYSTEMS / d) MANUAL STORES.
5. What do you do with the data that you receive from the other a) POSITIONS / b) DEPARTMENTS / c) SYSTEMS / d) MANUAL STORES.
6. Where do you store the data that you receive from the other a) POSITIONS / b) DEPARTMENTS / c) SYSTEMS / d) MANUAL STORES.
7. Who can access the data that you receive from the other a) POSITIONS / b) DEPARTMENTS / c) SYSTEMS / d) MANUAL STORES.
8. Why do you need the data that you receive from the various a) POSITIONS / b) DEPARTMENTS / c) SYSTEMS / d) MANUAL STORES.

Note: There will be four different sets of forms, showing:

1. The interaction with Positions
2. The interaction with Departments
3. The interaction with Systems
4. The interaction with Manual stores

There is likely to be duplication in this exercise, however once there is enough time then this should be a good method of checking that the information received is correct.

Communication Between a) POSITIONS / b) DEPARTMENTS / c) SYSTEMS / d) MANUAL STORES



Note:

The use of the symbols should help the users and developers to see what the system looks like form a graphical point of view. For example each different symbol represents a different aspect of the overall system which is shown in the key. This should also help the users to form a clear distinction between manual and automated transactions.

The diagrams can be used to see the level of automation currently in place

REFERENCE PAGES

IDENTIFICATION CODES

D = DEPARTMENT (2 DIGIT CODE) D + INDIVIDUAL NUMBER
P = POSITION (4 DIGIT CODE) P + DEPARTMENT NUMBER + DECIMAL POINT + INDIVIDUAL NUMBER
S = SYSTEM (2 DIGIT CODE) S + INDIVIDUAL NUMBER
M = MANUAL STORES (2 DIGIT CODE) M + INDIVIDUAL NUMBER

EXAMPLE OF THE USE OF CODES

(Each area could use its own page, this will depend on the number of codes needed for each section)

DEPARTMENT CODES

DEPARTMENT	CODE
PLANNING	D1
PRODUCTION	D2
ADMINISTRATION	D3
ACCOUNTING	D4
ENGINEERING	D5

POSITION CODES

POSITION	CODE
PLANNING MANAGER	P1.1
PRODUCTION MANAGER	P2.1
SHOP FLOOR MANAGER	P2.2
SECRETARY	P3.1
TYPIST	P3.2
FINANCE DIRECTOR	P4.1
BOOK KEEPER	P4.2
TECHNICAL MANAGER	P5.1
DRAWING SUPERVISOR	P5.2

SYSTEM CODES

SYSTEM	CODE
MRP	S1
OPAC	S2
OMAC	S3
SYM	S4
GYM	S5

MANUAL STORE CODES

MANUAL STORE	CODE
SALES ORDER BOOK ANALYSIS	M1
SHIFT MANAGER FORMS	M2
SHORT INTERVAL SHEETS	M3
FAULT LOG BOOK	M4

DATA DESCRIPTION SHEETS

DATA DESCRIPTION	DATA CODE
JOB RECIPE	DATA1
	DATA2
	DATA3
	DATA4
	DATA5
	DATA6
	DATA7
	DATA8
	DATA9
	DATA10
	DATA11
	DATA12
	DATA13
	DATA14
	DATA 15
	DATA 16

REFERENCE TABLES

……………………..CODES

…………………………………..	CODE
	1
	2
	3
	4
	5
	6
	7
	8
	9
	10
	11
	12
	13
	14
	15
	16
	17
	18
	19
	20
	21
	22
	23

URCAM DEVELOPMENT FORMS

FORM 1 HELP SHEET

What is the name of your

Write the name of the Position, Department, System or Manual store that you represent (in this process) in the relevant box.
FOR EXAMPLE
If you represent the PLANNING DEPARTMENT write PLANNING in the department box, as shown.

POSITION

DEPARTMENT — PLANNING

SYSTEM

MANUAL STORE

IDENTIFICATION CODE

(Codes are shown on the reference tables)

D1

Write your identification code in this box. You will find it on the reference pages.

FOR EXAMPLE
If you look at the DEPARTMENT TABLE and you represent the planning department, next to the name of your department will be a code in the identification box.

DEPARTMENT	CODE
PLANNING	D1
BUYING	D2

FORM 1

What is the name of your

POSITION

DEPARTMENT

SYSTEM

MANUAL STORE

IDENTIFICATION CODE

(Codes are shown on the reference tables)

....

<u>DEPARTMENT FORM 2 HELP SHEET</u>

Name of the other Departments that you communicate with ?

Your Identification Code.........D1.........

Write your identification code in this space. This will be found on form 1
FOR EXAMPLE

D1	✓ D2 BUYING	✓ D3 ADMIN	✓ D4 SALES	D5
D6	D7	D8	D9	D10

Tick the boxes of the departments that you communicate with in the relevant boxes.

FOR EXAMPLE

Use the reference tables to identify the departments that you communicate with.

DEPARTMENT	CODE
PLANNING	D1
BUYING	D2
ADMIN	D3
SALES	D4

………………………………FORM 2

Name of the other ……………… that you communicate with ?

Please fill in your identification code here

Your Identification Code……………………

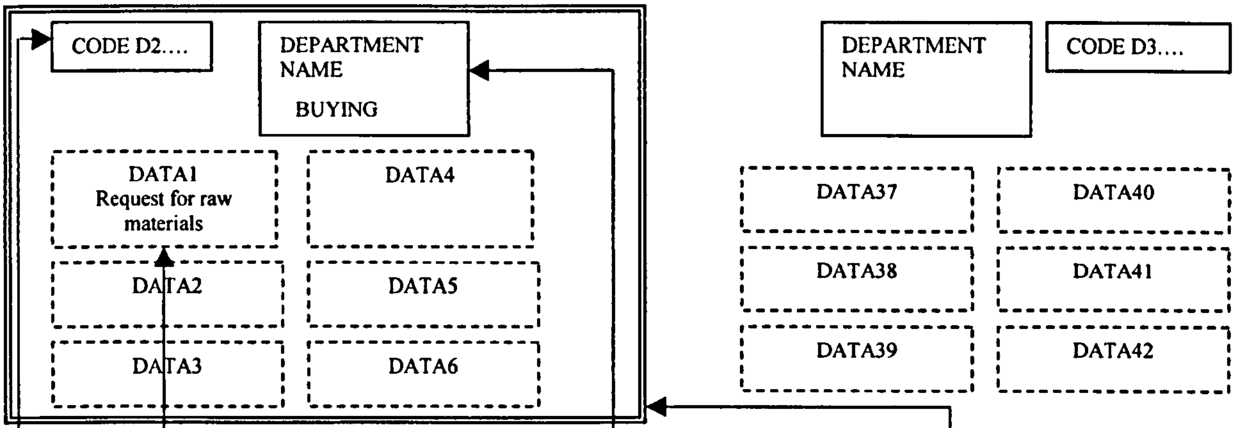
Please tick the boxes that represent the departments that you communicate with

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25
26	27	28	29	30

DEPARTMENT FORM 3 HELP SHEET

What data do you send to other departments?

Your Identification Code.........................



Write a very brief description of each piece of data that you send to each department.

FOR EXAMPLE

If you send Requests for raw materials to Buying (D2) fill in one box with that information as shown.

This area of data will be connected to one code

Write the code names of the departments that you send data to in the CODE boxes, OR fill in the department name in the DEPARTMENT NAME box.

FOR EXAMPLE

Use the reference tables to identify the departments that you communicate with.

DEPARTMENT	CODE
PLANNING	D1
BUYING	D2
ADMIN	D3
SALES	D4

............................FORM 3

What data do you send to other ....?

Your Identification Code........................

CODE....

NAME ....

DATA1

DATA4

DATA2

DATA5

DATA3

DATA6

NAME ....

CODE....

DATA19

DATA22

DATA20

DATA23

DATA21

DATA24

CODE....

NAME ....

DATA7

DATA10

DATA8

DATA11

DATA9

DATA12

NAME ....

CODE....

DATA25

DATA28

DATA26

DATA29

DATA27

DATA30

CODE....

NAME ....

DATA13

DATA16

DATA14

DATA17

DATA15

DATA18

NAME ....

CODE....

DATA31

DATA34

DATA32

DATA35

DATA33

DATA36

.... FORM 3

What data do you send to other ....?

Your Identification Code.......................

CODE.... NAME ....

DATA37 DATA40

DATA38 DATA41

DATA39 DATA42

CODE.... NAME ....

DATA43 DATA46

DATA44 DATA47

DATA45 DATA48

CODE.... NAME ....

DATA49 DATA52

DATA50 DATA53

DATA51 DATA54

NAME .... CODE....

DATA55 DATA58

DATA56 DATA59

DATA57 DATA60

NAME .... CODE....

DATA61 DATA64

DATA62 DATA65

DATA63 DATA66

NAME .... CODE....

DATA67 DATA70

DATA68 DATA71

DATA69 DATA72

.... FORM 3

What data do you send to other ....?

Your Identification Code........................

CODE.... NAME ....

DATA73 DATA76

DATA74 DATA77

DATA75 DATA78

CODE.... NAME ....

DATA79 DATA82

DATA80 DATA83

DATA81 DATA84

CODE.... NAME ....

DATA85 DATA88

DATA86 DATA89

DATA87 DATA90

NAME .... CODE....

DATA91 DATA94

DATA92 DATA95

DATA93 DATA96

NAME .... CODE....

DATA97 DATA100

DATA98 DATA101

DATA99 DATA102

NAME .... CODE....

DATA103 DATA106

DATA104 DATA107

DATA105 DATA108

………………………… FORM 3

What data do you send to other ………………………?

Your Identification Code……………………

CODE....	NAME …………………			NAME …………………	CODE....
DATA109	DATA112			DATA127	DATA130
DATA110	DATA113			DATA128	DATA131
DATA111	DATA114			DATA129	DATA132
CODE....	NAME …………………			NAME …………………	CODE....
DATA115	DATA118			DATA133	DATA136
DATA116	DATA119			DATA134	DATA137
DATA117	DATA120			DATA135	DATA138
CODE....	NAME …………………			NAME …………………	CODE....
DATA121	DATA124			DATA139	DATA142
DATA122	DATA125			DATA140	DATA143
DATA123	DATA126			DATA141	DATA144

DEPARTMENT FORM 4 HELP SHEET

What data do you receive from other departments?

Your Identification Code……………………

CODE D2….

DEPARTMENT NAME

DATA1
Buying

DATA4

DATA2

DATA5

DATA3

DATA6

DEPARTMENT NAME

CODE D3….

DATA37

DATA40

DATA42

DATA38

DATA39

Write a very brief description of each piece of data that you receive from each department.

FOR EXAMPLE
If you receive Buying quotas from Buying (D2) fill in one box with that information as shown.

This area of data will be connected to one code

Write the code names of the departments that you receive data from in the CODE boxes, OR fill in the department name in the DEPARTMENT NAME box.

FOR EXAMPLE

Use the reference tables to identify the departments that you communicate with.

DEPARTMENT	CODE
PLANNING	D1
BUYING	D2
ADMIN	D3
SALES	D4

..............................FORM 4

What data do you receive from other ....?

Your Identification Code........................

CODE....	NAME			NAME	CODE....
DATA1	DATA4			DATA19	DATA22
DATA2	DATA5			DATA20	DATA23
DATA3	DATA6			DATA21	DATA24
CODE....	NAME			NAME	CODE....
DATA7	DATA10			DATA25	DATA28
DATA8	DATA11			DATA26	DATA29
DATA9	DATA12			DATA27	DATA30
CODE....	NAME			NAME	CODE....
DATA13	DATA16			DATA31	DATA34
DATA14	DATA17			DATA32	DATA35
DATA15	DATA18			DATA33	DATA36

............................FORM 4

What data do you receive from other ....?

Your Identification Code........................

CODE....	NAME ....
DATA37	DATA40
DATA38	DATA41
DATA39	DATA42

NAME	CODE....
DATA55	DATA58
DATA56	DATA59
DATA57	DATA60

CODE....	NAME ....
DATA43	DATA46
DATA44	DATA47
DATA45	DATA48

NAME	CODE....
DATA61	DATA64
DATA62	DATA65
DATA63	DATA66

CODE....	NAME ....
DATA49	DATA52
DATA50	DATA53
DATA51	DATA54

NAME	CODE....
DATA67	DATA70
DATA68	DATA71
DATA69	DATA72

.... FORM 4

What data do you receive from other ....?

Your Identification Code........................

CODE.... | NAME ....

DATA73 | DATA76
DATA74 | DATA77
DATA75 | DATA78

CODE.... | NAME ....

DATA79 | DATA82
DATA80 | DATA83
DATA81 | DATA84

CODE.... | NAME ....

DATA85 | DATA88
DATA86 | DATA89
DATA87 | DATA90

NAME .... | CODE....

DATA91 | DATA94
DATA92 | DATA95
DATA93 | DATA96

NAME .... | CODE....

DATA97 | DATA100
DATA98 | DATA101
DATA99 | DATA102

NAME .... | CODE....

DATA103 | DATA106
DATA104 | DATA107
DATA105 | DATA108

..............................FORM 4

What data do you receive from other ....?

Your Identification Code........................

CODE....	NAME ....

DATA109	DATA112
DATA110	DATA113
DATA111	DATA114

NAME	CODE....

DATA127	DATA130
DATA128	DATA131
DATA129	DATA132

CODE....	NAME ....

DATA115	DATA118
DATA116	DATA119
DATA117	DATA120

NAME	CODE....

DATA133	DATA136
DATA134	DATA137
DATA135	DATA138

CODE....	NAME ....

DATA121	DATA124
DATA122	DATA125
DATA123	DATA126

NAME	CODE....

DATA139	DATA142
DATA140	DATA143
DATA141	DATA144

DEPARTMENT FORM 5 HELP SHEET

What do you do with the data that you receive from other departments?

Your Identification Code.........................

DATA CODE 1.......
DEPT. CODE...D2
DEPT. NAME
...BUYING............

Used to work out what we can ask for

DATA CODE.......
DEPT. CODE...
DEPT. NAME
....

Write what you do with the data that you receive in this box.

FOR EXAMPLE
If you use this data to work out what you can ask for fill in as shown.

Look at form 4 to see the data that you receive. For each piece of data, fill in the boxes with the following information.

EXAMPLE

If form 4 shows

DEPARTMENT FORM 4

What data do you receive from other departments?

Your Identification Code.........................

CODE D2....

DEPARTMENT NAME
......BUYING..........

CODE....

DATA1
Buying quotas

DATA4

DATA37

DATA40

DATA2
Daily expenditure

DATA5

DATA38

DATA41

DATA3

DATA6

DATA39

DATA42

..............................FORM 5

What do you do with the data that you receive from other ....?

Your Identification Code........................

DATA CODE....... CODE... .NAME		DATA CODE....... CODE... NAME	
DATA CODE....... CODE... NAME		DATA CODE....... .CODE... .NAME	
DATA CODE....... .CODE... NAME		DATA CODE....... CODE... NAME	
DATA CODE....... CODE... NAME		DATA CODE....... CODE... NAME	
DATA CODE....... CODE... NAME		DATA CODE....... .CODE... .NAME	
DATA CODE....... CODE... NAME		DATA CODE....... CODE... .NAME	
DATA CODE....... .CODE... NAME		DATA CODE....... CODE... NAME	
DATA CODE....... CODE... NAME		DATA CODE....... .CODE... .NAME	
DATA CODE....... CODE... . NAME		DATA CODE....... CODE... .NAME	
DATA CODE....... CODE... NAME		DATA CODE....... .CODE... NAME	
DATA CODE....... CODE... NAME		DATA CODE....... CODE... NAME	

.............................FORM 5

What do you do with the data that you receive from other ....?

Your Identification Code.......................

DATA CODE....... CODE... .NAME		DATA CODE....... CODE... NAME	
DATA CODE....... CODE... NAME		DATA CODE....... .CODE... .NAME	
DATA CODE....... .CODE... NAME		DATA CODE....... CODE... NAME	
DATA CODE....... CODE... NAME		DATA CODE....... CODE... NAME	
DATA CODE....... CODE... NAME		DATA CODE....... .CODE... .NAME	
DATA CODE....... CODE... NAME		DATA CODE....... CODE... .NAME	
DATA CODE....... .CODE... NAME		DATA CODE....... CODE... NAME	
DATA CODE....... CODE... NAME		DATA CODE....... .CODE... .NAME	
DATA CODE....... CODE... . NAME		DATA CODE....... CODE... .NAME	
DATA CODE....... CODE... NAME		DATA CODE....... .CODE... NAME	
DATA CODE....... CODE... NAME		DATA CODE....... CODE... NAME	

DEPARTMENT FORM 6 HELP SHEET

Where do you store the data that you receive from other departments?

Your Identification Code........................

DATA CODE 1....... DEPT. CODE...D2 DEPT. NAME ......BUYING..........	In the quotas cabinet	DATA CODE....... DEPT. CODE... DEPT. NAME	

Write where you store the data that you receive in this box.

FOR EXAMPLE
If you store the data in the quotas cabinet fill in as shown.

Look at form 4 to see the data that you receive. For each piece of data, fill in the boxes with the following information.

EXAMPLE

If form 4 shows

DEPARTMENT FORM 4

What data do you receive from other departments?

Your Identification Code........................

CODE D2....

DEPARTMENT NAME ......BUYING..........

DATA1 Buying quotas	DATA4
DATA2 Daily expenditure	DATA5
DATA3	DATA6

DEPARTMENT NAME ......BUYING..........

CODE....

DATA37	DATA40
DATA38	DATA41
DATA39	DATA42

............................FORM 6

Where do you store the data that you receive from other ....?

Your Identification Code........................

DATA CODE....... CODE... .NAME		DATA CODE....... CODE... NAME	
DATA CODE....... CODE... NAME		DATA CODE....... .CODE... .NAME	
DATA CODE....... .CODE... NAME		DATA CODE....... CODE... NAME	
DATA CODE....... CODE... NAME		DATA CODE....... CODE... NAME	
DATA CODE....... CODE... NAME		DATA CODE....... .CODE... .NAME	
DATA CODE....... CODE... NAME		DATA CODE....... CODE... .NAME	
DATA CODE....... .CODE... NAME		DATA CODE....... CODE... NAME	
DATA CODE....... CODE... NAME		DATA CODE....... .CODE... .NAME	
DATA CODE....... CODE... . NAME		DATA CODE....... CODE... .NAME	
DATA CODE....... CODE... NAME		DATA CODE....... .CODE... NAME	
DATA CODE....... CODE... NAME		DATA CODE....... CODE... NAME	

..............................FORM 6

Where do you store the data that you receive from other ....?

Your Identification Code........................

DATA CODE....... CODE... .NAME		DATA CODE....... CODE... NAME	
DATA CODE....... CODE... NAME		DATA CODE....... .CODE... .NAME	
DATA CODE....... .CODE... NAME		DATA CODE....... CODE... NAME	
DATA CODE....... CODE... NAME		DATA CODE....... CODE... NAME	
DATA CODE....... CODE... NAME		DATA CODE....... .CODE... .NAME	
DATA CODE....... CODE... NAME		DATA CODE....... CODE... .NAME	
DATA CODE....... .CODE... NAME		DATA CODE....... CODE... NAME	
DATA CODE....... CODE... NAME		DATA CODE....... .CODE... .NAME	
DATA CODE....... CODE... . NAME		DATA CODE....... CODE... .NAME	
DATA CODE....... CODE... NAME		DATA CODE....... .CODE... NAME	
DATA CODE....... CODE... NAME		DATA CODE....... CODE... NAME	

DEPARTMENT FORM 7 HELP SHEET

Who can access the data that you receive from other departments?

Your Identification Code........................

DATA CODE 1....... DEPT. CODE...D2 DEPT. NAME ...BUYING................	Planner and the Sales Department	DATA CODE....... DEPT. CODE... DEPT. NAME	

Write who can access the data that you receive in this box.

FOR EXAMPLE
If the planner and the sales department can access the data, fill in as shown.

Look at form 4 to see the data that you receive. For each piece of data, fill in the boxes with the following information.

EXAMPLE

If form 4 shows

DEPARTMENT FORM 4

What data do you receive from other departments?

Your Identification Code........................

CODE D2....

DEPARTMENT NAME ......BUYING..........

DATA1 Buying quotas	DATA4
DATA2 Daily expenditure	DATA5
DATA3	DATA6

DEPARTMENT NAME ......BUYING..........

CODE....

DATA37	DATA40
DATA38	DATA41
DATA39	DATA42

..............................FORM 7

Who can access the data that you receive from other ....?

Your Identification Code........................

DATA CODE....... CODE... .NAME		DATA CODE....... CODE... NAME	
DATA CODE....... CODE... NAME		DATA CODE....... .CODE... .NAME	
DATA CODE....... .CODE... NAME		DATA CODE....... CODE... NAME	
DATA CODE....... CODE... NAME		DATA CODE....... CODE... NAME	
DATA CODE....... CODE... NAME		DATA CODE....... .CODE... .NAME	
DATA CODE....... CODE... NAME		DATA CODE....... CODE... .NAME	
DATA CODE....... .CODE... NAME		DATA CODE....... CODE... NAME	
DATA CODE....... CODE... NAME		DATA CODE....... .CODE... .NAME	
DATA CODE....... CODE... . NAME		DATA CODE....... CODE... .NAME	
DATA CODE....... CODE... NAME		DATA CODE....... .CODE... NAME	
DATA CODE....... CODE... NAME		DATA CODE....... CODE... NAME	

..............................FORM 7

Who can access the data that you receive from other ....?

Your Identification Code.........................

DATA CODE....... CODE... .NAME		DATA CODE....... CODE... NAME	
DATA CODE....... CODE... NAME		DATA CODE....... .CODE... .NAME	
DATA CODE....... .CODE... NAME		DATA CODE....... CODE... NAME	
DATA CODE....... CODE... NAME		DATA CODE....... CODE... NAME	
DATA CODE....... CODE... NAME		DATA CODE....... .CODE... .NAME	
DATA CODE....... CODE... NAME		DATA CODE....... CODE... .NAME	
DATA CODE....... .CODE... NAME		DATA CODE....... CODE... NAME	
DATA CODE....... CODE... NAME		DATA CODE....... .CODE... .NAME	
DATA CODE....... CODE... . NAME		DATA CODE....... CODE... .NAME	
DATA CODE....... CODE... NAME		DATA CODE....... .CODE... NAME	
DATA CODE....... CODE... NAME		DATA CODE....... CODE... NAME	

DEPARTMENT FORM 8 HELP SHEET

Why do you need the data that you receive from other departments?

Your Identification Code.......................

DATA CODE 1.......
DEPT. CODE...D2
DEPT. NAME
.......BUYING............

To ensure that we keep
track of the expenditure in

DATA CODE.......
DEPT. CODE...

Write why you need the data that you receive in this box.

FOR EXAMPLE
If you use this data to work out what you can ask for fill in as shown.

Look at form 4 to see the data that you receive. For each piece of data, fill in the boxes with the following information.

EXAMPLE

If form 4 shows

DEPARTMENT FORM 4

What data do you receive from other departments?

Your Identification Code.......................

CODE D2....

DEPARTMENT
NAME
......BUYING..........

DATA1
Buying quotas

DATA4

DATA2
Daily expenditure

DATA5

DATA3

DATA6

DEPARTMENT
NAME
......BUYING..........

CODE....

DATA37

DATA40

DATA38

DATA41

DATA39

DATA42

..............................FORM 8

Why do you need the data that you receive from other ....?

Your Identification Code.........................

DATA CODE....... CODE... .NAME		DATA CODE....... CODE... NAME	
DATA CODE....... CODE... NAME		DATA CODE....... .CODE... .NAME	
DATA CODE....... .CODE... NAME		DATA CODE....... CODE... NAME	
DATA CODE....... CODE... NAME		DATA CODE....... CODE... NAME	
DATA CODE....... CODE... NAME		DATA CODE....... .CODE... .NAME	
DATA CODE....... CODE... NAME		DATA CODE....... CODE... .NAME	
DATA CODE....... .CODE... NAME		DATA CODE....... CODE... NAME	
DATA CODE....... CODE... NAME		DATA CODE....... .CODE... .NAME	
DATA CODE....... CODE... . NAME		DATA CODE....... CODE... .NAME	
DATA CODE....... CODE... NAME		DATA CODE....... .CODE... NAME	
DATA CODE....... CODE... NAME		DATA CODE....... CODE... NAME	

...............................FORM 8

Why do you need the data that you receive from other ....?

Your Identification Code.........................

DATA CODE....... CODE... .NAME		DATA CODE....... CODE... NAME	
DATA CODE....... CODE... NAME		DATA CODE....... .CODE... .NAME	
DATA CODE....... .CODE... NAME		DATA CODE....... CODE... NAME	
DATA CODE....... CODE... NAME		DATA CODE....... CODE... NAME	
DATA CODE....... CODE... NAME		DATA CODE....... .CODE... .NAME	
DATA CODE....... CODE... NAME		DATA CODE....... CODE... .NAME	
DATA CODE....... .CODE... NAME		DATA CODE....... CODE... NAME	
DATA CODE....... CODE... NAME		DATA CODE....... .CODE... .NAME	
DATA CODE....... CODE... . NAME		DATA CODE....... CODE... .NAME	
DATA CODE....... CODE... NAME		DATA CODE....... .CODE... NAME	
DATA CODE....... CODE... NAME		DATA CODE....... CODE... NAME	

APPENDIX E
USER QUESTIONNAIRE
(EVALUATION OF URCAM)

QUESTIONNAIRE FOR USERS

This questionnaire is aimed at users of information systems (from shop floor to management levels). The questions are used to form an evaluation of a proposed method, which is based on the use of development forms. These development forms are used as templates for developers to capture information from users about their day-to-day activities.

The evaluation process requires the users to fill in a set of forms that are presented to them individually or in a group. The users will fill in these forms using the help of the developer and the help sheets that are provided. Each form contains one question about the users day-to-day activities (for example asking questions such as what information do you send to other departments? where do you store the information?). There are 8 different questions that are contained on 8 different forms.

Once the forms are filled in, this questionnaire is presented to the user, who will be required to answer questions relating to:

1. their previous experience of taking part in this exercise
2. the usability of the forms,
3. the effectiveness of the information that the users provided,
4. the usefulness of the help sheets,
5. their comfort with filling in the forms and
6. the effect that the forms might have on user involvement.

The last area of the questionnaire (comments) gives the user the opportunity to give their opinions on any aspects of the forms that they felt are not covered in the questionnaire.

Note:

- The interviewer will be with the user when they are filling in the forms and the questionnaire.
- As soon as the forms are filled in the users will complete the questionnaire.
- The forms will be laid down in front of the user so that they can refer back to them at any time.

Name:		
Job title		
Contact details:	Phone Number:	
	Fax Number:	
	Email:	

Please tick the boxes or circles that represent your answers and write on the dotted lines where necessary.

Previous experience

1. Have you been asked questions about your day-to-day activities before?

   Yes ☐ No ☐

   If yes continue with question 2, If no go to question 5

2. What method(s) were used to collect the information regarding your day-to-day job activities?

   ....

   ....

3. How confident did you feel that the information that you gave when you were asked about your day-to-day activities would be used in developing the new system?

   Not at all   Confident   Very Confident

   1 ○   2 ○   3 ○   4 ○   5 ○

4. How much of what you expected to go into the system was actually in the system?

   None   Some   All

   1 ○   2 ○   3 ○   4 ○   5 ○

Usability of the forms

5. Please write down the good and bad points that you felt about each form that you filled in

   Example of a Good Point: "I understood the question"
   Example of a Bad Point: "The boxes were too small"

Form Number	Good Points	Bad Points
1		
2		
3		
4		
5		
6		
7		
8		

Effectiveness of information gained from using the forms

6. How does the information that you have given on the forms compare with the information you would have given verbally?

   Worse — The same — Better

   1 ○ 2 ○ 3 ○ 4 ○ 5 ○

   Why?..

7. How confident are you that the information that you have given is going to be used because you have written down your requirements?

   Not at all — Confident — Very Confident

   1 ○ 2 ○ 3 ○ 4 ○ 5 ○

   Why?..

8. Was the time required to complete the forms

Too short?		Enough?		Too long?
1 ○	2 ○	3 ○	4 ○	5 ○

Usage of help sheets

9. How well did you understand the help sheets?

Not at all		Reasonably		Totally
1 ○	2 ○	3 ○	4 ○	5 ○

10. How useful did you find the help sheets?

Not at all		Reasonably		Very
1 ○	2 ○	3 ○	4 ○	5 ○

Comfort with filling in forms

11. To what extent do you fear writing down information on the forms?

No fear		Cautious		Very Fearful
1 ○	2 ○	3 ○	4 ○	5 ○

If you ticked a circle numbered between 2 and 5 was the reason

(Please tick one box only)

You are not confident with your spelling? ☐
You are not confident with your grammar? ☐
You find it hard to remember what you do and write it down? ☐
You think that your handwriting is not easy to understand? ☐

If there is another reason, please write it down
here……………………………………………………………………..

12. To what extent do you fear filling in your name on the forms?

No fear 1 ◯ 2 ◯ | Cautious 3 ◯ 4 ◯ | Very Fearful 5 ◯

If you ticked a circle numbered between 2 and 5 was the reason

(Please tick one box only)

You feel that if you make a mistake it will effect your job in some way? ☐

You do not want to take responsibility for the information that is used to develop the system? ☐

You do not want to be asked any questions about what you have written at a later stage? ☐

If there is another reason, please write it down here…………………………………………………………………..

Effects on involvement

13. How likely are you to use the information system because you have been involved in filling in the forms?

Less 1 ◯ 2 ◯ | No difference 3 ◯ 4 ◯ | More 5 ◯

14. How involved does filling in the forms make you feel?

Less 1 ◯ 2 ◯ | No difference 3 ◯ 4 ◯ | More 5 ◯

15. How confident do you feel after you have filled in the forms that what you have said will be used to produce the new system?

Less		No difference		More
1 ○	2 ○	3 ○	4 ○	5 ○

If you have any other comments about the forms please use this area to write them down (For example comments about the layout of the forms, the number of questions, the time taken or anything else that you have not been able to say in the questionnaire)

....
....
....
....
....
....

APPENDIX F
DEVELOPER QUESTIONNAIRE
(EVALUATION OF URCAM)

QUESTIONNAIRE FOR DEVELOPERS

This questionnaire is aimed at developers of information systems. The questions are used to form an evaluation of a proposed method, which is based on the use of development forms. These development forms are used as templates for developers to capture information from users about their day-to-day activities.

The evaluation process requires the developer to go through the forms and evaluate them according to their experience taking users through the development process. The method is based on users filling in a set of forms that are presented to them individually or in a group. The users will fill in these forms using the help of the developer and the help sheets that are provided. Each form contains one question about the users day-to-day activities (for example asking questions such as what information do you send to other departments? where do you store the information?). There are 8 different questions that are contained on 8 different forms.

Once the forms and the process is explained to the developer, they will be required to answer questions relating to users and :

1. their previous experience of taking part in this exercise
2. the usability of the forms,
3. the effectiveness of the information that the users would provide,
4. the usefulness of the help sheets,
5. their comfort with filling in the forms and
6. the effect that the forms might have on user involvement.

The last area of the questionnaire (comments) gives the user the opportunity to give their opinions on any aspects of the forms that they felt are not covered in the questionnaire.

Note:

- The interviewer will be with the developer while they are evaluating the forms.
- As soon as the forms are evaluated the developer will complete the questionnaire.
- The forms will be laid down in front of the developer so that they can refer back to them at any time.

Name:		
Job title		
Contact details:	Phone Number:	
	Fax Number:	
	Email:	

Please tick the boxes or circles that represent your answers and write on the dotted lines where necessary.

Previous experience

1. Have the users been asked questions about your day-to-day activities before?

   Yes ☐ No ☐

   If yes continue with question 2, If no go to question 5

2. What method(s) were used to collect the information regarding their day-to-day job activities?

   ....

   ....

3. How confident do you think that they would feel that the information that they gave when they were asked about their day-to-day activities would be used in developing the new system?

   Not at all / Confident / Very Confident
   1 ○ 2 ○ 3 ○ 4 ○ 5 ○

4. How much of what they expected to go into the system was actually in the system?

   None / Some / All
   1 ○ 2 ○ 3 ○ 4 ○ 5 ○

Usability of the forms

5. Please write down the good and bad points that you felt about each form

   Example of a Good Point: "I understood the question"
   Example of a Bad Point: "The boxes were too small"

Form Number	Good Points	Bad Points
1		
2		
3		
4		
5		
6		
7		
8		

Effectiveness of information gained from using the forms

6. How does the information that the users would have given on the forms compare with the information they would have given verbally?

   Worse / The same / Better
   1 ◯ 2 ◯ 3 ◯ 4 ◯ 5 ◯

   Why?..

7. How confident are you that the information that they would have given is going to be used because they have written down their requirements?

   Not at all / Confident / Very Confident
   1 ◯ 2 ◯ 3 ◯ 4 ◯ 5 ◯

   Why?..

8. Do you think that the time required to complete the forms was

Too short?		Enough?		Too long?
1 ◯	2 ◯	3 ◯	4 ◯	5 ◯

Usage of help sheets

9. How well did you think that the users would understand the help sheets?

Not at all		Reasonably		Totally
1 ◯	2 ◯	3 ◯	4 ◯	5 ◯

10. How useful did you think the help sheets would be?

Not at all		Reasonably		Very
1 ◯	2 ◯	3 ◯	4 ◯	5 ◯

Comfort with filling in forms

11. To what extent do you think that users would fear writing down information on the forms?

No fear		Cautious		Very Fearful
1 ◯	2 ◯	3 ◯	4 ◯	5 ◯

If you ticked a circle numbered between 2 and 5 was the reason

(Please tick one box only)

They are not confident with your spelling? ☐
They are not confident with your grammar? ☐
They find it hard to remember what you do and write it down? ☐
They think that your handwriting is not easy to understand? ☐

If there is another reason, please write it down here………………………………………………………………

12. To what extent do you think that they would fear filling in their name on the forms?

No fear		Cautious		Very Fearful
1 ◯	2 ◯	3 ◯	4 ◯	5 ◯

If you ticked a circle numbered between 2 and 5 was the reason

(Please tick one box only)

They feel that if you make a mistake it will effect your job in some way?	☐
They do not want to take responsibility for the information that is used to develop the system?	☐
They do not want to be asked any questions about what you have written at a later stage?	☐

If there is another reason, please write it down here...

<u>Effects on involvement</u>

13. How likely do you think that users are to actually use the information system because they have been involved in filling in the forms?

Less		No difference		More
1 ◯	2 ◯	3 ◯	4 ◯	5 ◯

14. How involved do you think that filling in the forms make them feel?

Less		No difference		More
1 ◯	2 ◯	3 ◯	4 ◯	5 ◯

15. How confident do you think that they would feel after they have filled in the forms that what they have said will be used to produce the new system?

Less		No difference		More
1 ○	2 ○	3 ○	4 ○	5 ○

If you have any other comments about the forms please use this area to write them down (For example comments about the layout of the forms, the number of questions, the time taken or anything else that you have not been able to say in the questionnaire)

....
....
....
....
....
....