HILARY BERGER
BSc (Hons) Business Information Systems

The Utility of Using a RAD-type Development Approach for a
Large, Complex Information System

A thesis submitted for the degree of
Doctor of Philosophy

Information Systems, Business School
University of Wales Institute, Cardiff

July 2005

Director of Studies
Internal Supervisor
External Supervisor

Pat Cleary
Kelvin Hughes
Paul Beynon-Davies
DECLARATION

I hereby declare that this dissertation, submitted in partial fulfilment of the requirements for the degree of Doctorate of Philosophy and entitled "The utility of using a RAD-type development approach for a large, complex information system" represents my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material.

Signed ........................................... (Candidate)

Date ..........4/10/05.................

STATEMENT 1

This thesis has not already been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed ........................................... (Candidate)

Date ..........4/10/05.................

STATEMENT 2

I hereby give consent for my thesis to be available for photocopying and for inter library loan, and for the title and summary to be made available to outside organizations.

Signed ........................................... (Candidate)

Date ..........4/10/05.................
ACKNOWLEDGMENTS

I am indebted to my internal supervisory committee, Pat Cleary and Kelvin Hughes, for their helpful and insightful comments together with their guidance and reassurance during this study.

A very special thanks goes to Paul Beynon-Davies, external supervisor, for giving me the support to explore, define and achieve my doctoral research project and without whom this project would not have been possible.

Gratitude goes to Len Arthur who provided insight into the values of philosophy, social theory and ethnographic research.

I would also like to thank the Project Managers of the New IT System and all participants that were involved, both internal and outsourced, who shared their time and experience, and who co-operated fully in this research unreservedly.

Gratitude is further extended to The National Assembly of Wales who funded this PhD project. Finally appreciation goes to UWIC Business School for their financial support during my paper writing and conference presentations, and to the patient staff of the Library Resources Centre.
ABSTRACT

Rapid Application Development (RAD) is an iterative and incremental development approach that evolved to address problems associated with the more structured development approaches such as the Waterfall Model. Even though RAD is becoming an increasingly accepted development approach, much of the existing literature focuses on small to medium sized development projects. There is considerable debate about its application for large, complex IS development arenas.

This research utilises a development project currently being implemented within UK Regional Government as a case study. It represents an atypical opportunity to examine the utility of RAD within a large, complex IS development, presenting the real-life context, experiences and commentary from individuals directly involved.

An interpretive stance is adopted to gain a broad view of the organizational environment of the IS and the wider external context within which the information system is related. An ethnographic approach was selected enabling a richer and deeper interpretation, and a more comprehensive understanding of the issues under investigation. This methodology included non-participatory observation, indirect observation and informal semi-structured interviews. It also involved multiple strategies of data collection and analysis to facilitate cross-checking and to yield stronger substantiation of analysis.

The thesis examines the cultural aspects inherent in the studied environment that impacted severely upon the project. It further explores a number of other issues held problematic for large and complex development arenas: managing user involvement and expectations, communications, requirements elicitation, decision-making and testing. Analysis of the case study material has aided the production of a model of critical success factors that could be applied to other such environments adopting a RAD-type approach. Thus it contributes to the field of IS knowledge by informing the debate surrounding applicability of RAD across large, complex development arenas.
TABLES OF CONTENTS

CHAPTER 1. INTRODUCTION TO THE RESEARCH

1. Introduction
   1.1 RAD – Historical Overview
   1.2 Research Rationale
   1.3 Research Aims, Objectives and Questions
   1.4 Introduction to the Case Study
   1.5 Introduction to the Research Approach
   1.6 Thesis Contributions
   1.7 Organization of the Thesis

CHAPTER 2. RESEARCH DESIGN AND METHODOLOGY

2. Introduction
   2.1 Case Study Context and Research Approach Adopted
      2.1.1 Overview of the Case Study Context
      2.1.2 Overview of the Research Approach
   2.2 Research Philosophy - Phenomenology
   2.3 Research Design and Methodology
      2.3.1 Case Study Approach
      2.3.2 Suitability of the Approach to IS Research
   2.4 Data Collection and Research Techniques
      2.4.1 Direct Observation and Indirect Observation
      2.4.2 Informal Semi-structured Interviews
         2.4.2.1 Interview Population
      2.4.3 Shadowing of Key People
      2.4.4 Spontaneous Conversations
   2.5 Data Organization and Analysis Techniques
      2.5.1 Data Capture and Data Analysis
      2.5.2 Application of NVivo to the Research Case Study
   2.6 Reliability and Ecological Validation
      2.6.1 Role of the Researcher
      2.6.2 Access to the Project Environment
      2.6.3 Ecological Validity
      2.6.4 Triangulation
      2.6.5 Researcher Credibility
      2.6.6 Ethical Issues
   2.7 Research Schedule
   2.8 Strengths and Weakness of Chosen Methodology
   2.8 Summary

CHAPTER 3. RAD – A LITERATURE REVIEW

3. Introduction
   3.1 Brief History of the Development of Information Systems
### 3.1 Information Systems Development Themes

#### 3.2 Evolution of Development Environments
- 3.2.1 Monoliths to PCs
- 3.2.2 High Level Development Environments
- 3.2.3 CASE Tools

#### 3.3 Overview of System Development Methodologies
- 3.3.1 Waterfall Model
- 3.3.2 Structured Systems Analysis and Design Method (SSADM)

#### 3.4 Modular Development
- 3.4.1 Boehm’s Spiral Model
- 3.4.2 Rapid Prototyping
- 3.4.3 Evolutionary Prototyping

#### 3.5 What is RAD?
- 3.5.1 RAD Defined - The Essence of RAD
- 3.5.2 Types of RAD Project
- 3.5.3 RAD Stages
- 3.5.4 Characteristics of RAD
  - 3.5.4.1 JAD workshops
  - 3.5.4.2 MoSCoW, Timeboxing and Descoping
  - 3.5.4.3 Prototyping

#### 3.6 RAD and the DSDM Approach
- 3.7.1 RAD - A Genesis
- 3.7.2 Project Size, Type and Complexity
  - 3.7.2.1 RAD – Framework, Method or Rapid Tools/Techniques?
- 3.7.3 RAD – Quality Systems Development or Quick and Dirty Solution?
- 3.7.4 Implications of Culture on a RAD Development Process.
  - 3.7.4.1 RAD and Cultural Changes
  - 3.7.4.2 Culture and Stakeholders
  - 3.7.4.3 Blame Culture
  - 3.7.4.4 Team culture

#### 3.8 Project Factors of RAD Development

#### 3.9 Summary

---

### CHAPTER 4. RAD DEVELOPMENT FACTORS AND STAKEHOLDER ISSUES

#### 4. Introduction

#### 4.1 User Involvement
- 4.1.1 User Definition
- 4.1.2 User Involvement and User Participation
- 4.1.3 Right Users
- 4.1.4 User Expectation
- 4.1.5 Disadvantages of User Involvement

#### 4.2 Requirements Elicitation
- 4.2.1 Issues of Requirements Elicitation

#### 4.3 Decision-Making

#### 4.4 Communications
- 4.4.1 Visibility and Transparency of Communications
CHAPTER 6. RESEARCH THEMES

6. Introduction
6.1 Organizational Structure
   6.1.1 Rationality
   6.1.2 Bureaucratic Cultures
6.2 Culture
   6.2.1 Culture and a RAD-type Development Approach.
   6.2.2 Individual Culture
   6.2.3 Group Culture
   6.2.4 Organizational Culture
6.3 The Analysis Framework
   6.3.1 Relationships within the Research Data
   6.3.2 Presentation of Analysis
6.4 Summary

CHAPTER 7. ANALYSIS: USER INVOLVEMENT, USER EXPECTATION AND COMMUNICATIONS

7. Introduction
7.1 User Involvement
   7.1.1 Right Users
      7.1.1.1 Inflexibility of Business Analysts
      7.1.1.2 Exclusion of Former IT Personnel
      7.1.1.3 Testing Issues
   7.1.2 Availability of Users
   7.1.3 Allocation to the Project
   7.1.4 A Management View
   7.1.5 External Factors
   7.1.6 Conclusions
7.2 User Expectation
   7.2.1 Unrealistic Levels
   7.2.2 User Perceptions
   7.2.3 User Acceptance
   7.2.4 Training Issues
   7.2.5 Prototyping
      7.2.5.1 Prototyping – Model Office
   7.2.6 User Expectation – Conclusions
7.3 Communications
   7.3.1 Communications – Structure
   7.3.2 Communications – Poor and Ineffective
   7.3.3 Internal Communications
   7.3.4 External Communications
   7.3.5 Visibility and Transparency
   7.3.6 Shared Experiences
   7.3.7 Communications Gap
   7.3.8 Culpability and Justification
   7.3.9 Tacit, Explicit and Implicit Knowledge
   7.3.10 User Reception
CHAPTER 8. ANALYSIS: REQUIREMENTS ELICITATION, DECISION-MAKING AND TESTING

8. Introduction
8.1 Requirements Elicitation
  8.1.1 JAD Sessions, Timeboxing
  8.1.2 Cultural Constraints
  8.1.3 Scheme Complexity Issues
  8.1.4 Development Paradox
  8.1.5 Requirement Elicitation - Conclusions
8.2 Decision-making
  8.2.1 Impact of Bureaucratic Culture on Decision-Making
  8.2.2 Business Champions
  8.2.3 Stakeholder Co-operation / 'Fit for Purpose' Concept
  8.2.4 Decision-making - Conclusions
8.3 Testing – System Testing and UAT Testing
  8.3.1 UAT Testing
  8.3.2 Testing - Scheduling
  8.3.3 Joint Testing Environment
  8.3.4 Testing – Conclusions
8.4. Chapter Conclusions

CHAPTER 9. CONCLUSIONS

9. Introduction
9.1 Research Questions Addressed
  9.1.1 Research Question 1 - Which issues are considered to have the most affect upon a RAD-type development?
    9.1.1.1 Project Size and Complexity
    9.1.1.2 Issues of Quality
    9.1.1.3 Cultural Issues
    9.1.1.4 User Involvement, User Expectation and Communications
    9.1.1.5 Requirements Elicitation and Decision-Making
  9.1.2 Research Question 2 - What are the core features of a RAD-type development approach, and were they applied to the case study development project?
    9.1.2.1 Core Features of RAD
    9.1.2.2 Alignment of DSDM 9 Principles to the Case Study Project
  9.1.3 Research Question 3 : What are the major influencing factors that affected the development process of the research case study project?
    9.1.3.1 User Involvement
    9.1.3.2 Requirements Elicitation and User Expectation
    9.1.3.3 Decision-making
9.1.3.4 Communications

9.1.4 Research Question 4: Were the problems experienced attributable to the RAD-type development approach adopted, or to other aspects of the research case study project?

9.1.4.1 Problems Experienced – Aspects of a RAD-type Development Approach?

9.1.4.2 Problems Experienced - Aspects of the Case Study Development Project?

9.1.4.3 Model of Critical Success Factors

9.1.4.4 Utility of a RAD-type Approach

9.2 Constraints and Limitations of the Research Study

9.3 Researcher’s Integrity and Influence

9.4 Directions for Further Research

9.5 Conference Papers Presented and Published in Proceedings

REFERENCES
<table>
<thead>
<tr>
<th>Figure No.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td><em>Generic Process Model</em></td>
</tr>
<tr>
<td>3.1</td>
<td><em>SDLC Lifecycle</em></td>
</tr>
<tr>
<td>3.2</td>
<td><em>Waterfall Model</em></td>
</tr>
<tr>
<td>3.3</td>
<td><em>SSADM and the System Development Lifecycle</em></td>
</tr>
<tr>
<td>3.4</td>
<td><em>Modular Delivery</em></td>
</tr>
<tr>
<td>3.5</td>
<td><em>Incremental Delivery</em></td>
</tr>
<tr>
<td>3.6</td>
<td><em>Boehm's Spiral Model</em></td>
</tr>
<tr>
<td>3.7</td>
<td><em>Rapid Prototyping</em></td>
</tr>
<tr>
<td>3.8</td>
<td><em>The Waterfall Model and RAD Development Approaches</em></td>
</tr>
<tr>
<td>3.9</td>
<td><em>RAD within a Planned Infrastructure</em></td>
</tr>
<tr>
<td>3.10</td>
<td><em>The Four Essential Aspects of RAD</em></td>
</tr>
<tr>
<td>3.11</td>
<td><em>The Integration of Speed, Quality and Low Cost</em></td>
</tr>
<tr>
<td>3.12</td>
<td><em>Four Phases of RAD</em></td>
</tr>
<tr>
<td>3.13</td>
<td><em>Traditional SDLC aligned with RAD and JAD</em></td>
</tr>
<tr>
<td>3.14</td>
<td><em>RAD and an Iterative Prototyping Process</em></td>
</tr>
<tr>
<td>3.15</td>
<td><em>DSM Development Lifecycle</em></td>
</tr>
<tr>
<td>3.16</td>
<td><em>The Relationship between RAD and Quality of Systems</em></td>
</tr>
<tr>
<td>4.1</td>
<td><em>Audience needs aligned to Communication Channels</em></td>
</tr>
<tr>
<td>5.1</td>
<td><em>TIMELINE - Chronology of Case Study Project</em></td>
</tr>
<tr>
<td>5.2</td>
<td><em>Key Stakeholder of CAPM Division</em></td>
</tr>
<tr>
<td>5.3</td>
<td><em>CAP Scheme Relationship Diagram</em></td>
</tr>
<tr>
<td>5.4</td>
<td><em>Relationship Map of the SCPS CAP Scheme</em></td>
</tr>
<tr>
<td>5.5</td>
<td><em>The New IT System Project Reporting Structure</em></td>
</tr>
<tr>
<td>5.6</td>
<td><em>Project Structure</em></td>
</tr>
<tr>
<td>5.7</td>
<td><em>Generic Process Model</em></td>
</tr>
<tr>
<td>5.8</td>
<td><em>High-level Overview of New IT System</em></td>
</tr>
<tr>
<td>5.9</td>
<td><em>Workflow of Common Processes</em></td>
</tr>
<tr>
<td>5.10</td>
<td><em>Multi-Tier Distributed Model of New IT System</em></td>
</tr>
<tr>
<td>5.11</td>
<td><em>Illustration of the Physical Architecture</em></td>
</tr>
<tr>
<td>5.12</td>
<td><em>Overview of the IAF Approach</em></td>
</tr>
<tr>
<td>5.13</td>
<td><em>Key Mile Stone Plan</em></td>
</tr>
<tr>
<td>5.14</td>
<td><em>Amended Milestone Map showing External and Internal impacts on the development project</em></td>
</tr>
<tr>
<td>5.15</td>
<td><em>Pictorial Representation of Periods of Change</em></td>
</tr>
<tr>
<td>5.16</td>
<td><em>New IT Project Development Cycle</em></td>
</tr>
<tr>
<td>5.17</td>
<td><em>Business Rules Process</em></td>
</tr>
<tr>
<td>6.1</td>
<td><em>Levels of Uniqueness in Human Mental Programming</em></td>
</tr>
<tr>
<td>6.2</td>
<td><em>High-level View of the Research Journey</em></td>
</tr>
<tr>
<td>6.3</td>
<td><em>A Holistic View of the Factors Affecting the Case Study Project</em></td>
</tr>
</tbody>
</table>

**Page No.:**
- Figure 1.1: 28
- Figure 3.1: 73
- Figure 3.2: 74
- Figure 3.3: 76
- Figure 3.4: 79
- Figure 3.5: 79
- Figure 3.6: 80
- Figure 3.7: 82
- Figure 3.8: 88
- Figure 3.9: 88
- Figure 3.10: 89
- Figure 3.11: 90
- Figure 3.12: 91
- Figure 3.13: 94
- Figure 3.14: 96
- Figure 3.15: 98
- Figure 3.16: 110
- Figure 4.1: 136
- Figure 5.1: 145
- Figure 5.2: 149
- Figure 5.3: 153
- Figure 5.4: 154
- Figure 5.5: 157
- Figure 5.6: 161
- Figure 5.7: 163
- Figure 5.8: 165
- Figure 5.9: 166
- Figure 5.10: 168
- Figure 5.11: 169
- Figure 5.12: 174
- Figure 5.13: 178
- Figure 5.14: 179
- Figure 5.15: 181
- Figure 5.16: 182
- Figure 5.17: 186
- Figure 6.1: 205
- Figure 6.2: 212
- Figure 6.3: 213
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table No.</th>
<th>Description</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 2.1</td>
<td><em>Research Schedule</em></td>
<td>58</td>
</tr>
<tr>
<td>Table 3.1</td>
<td><em>DSDM 9 RAD Principles &amp; Martin’s RAD Philosophy</em></td>
<td>97</td>
</tr>
<tr>
<td>Table 3.2</td>
<td><em>RAD – A Table of Key Issues Affecting RAD Developments</em></td>
<td>121</td>
</tr>
<tr>
<td>Table 5.1</td>
<td><em>Programme of Change Projects</em></td>
<td>156</td>
</tr>
<tr>
<td>Table 5.2</td>
<td><em>Project Community</em></td>
<td>160</td>
</tr>
<tr>
<td>Table 5.3</td>
<td><em>Key Software Components of New IT System</em></td>
<td>170</td>
</tr>
<tr>
<td>Table 5.4</td>
<td><em>DSDM 9 Principles</em></td>
<td>175</td>
</tr>
<tr>
<td>Table 5.5</td>
<td><em>DSDM 9 Principles as Applied to the Case Study Project</em></td>
<td>175</td>
</tr>
<tr>
<td>Table 6.1</td>
<td><em>A Table of Key Issues Affecting RAD Developments</em></td>
<td>211</td>
</tr>
<tr>
<td>Table 9.1</td>
<td><em>The 9 DSDM Principles</em></td>
<td>315</td>
</tr>
<tr>
<td>Table 9.2</td>
<td><em>Alignment of the DSDM 9 Principles to the Case Study</em></td>
<td>316</td>
</tr>
<tr>
<td>Table 9.3</td>
<td><em>Relationships between the Research Themes and Development Factors</em></td>
<td>338</td>
</tr>
<tr>
<td>Table 9.4</td>
<td><em>RAD – An Model of Critical Success Factors affecting RAD Development Projects</em></td>
<td>340</td>
</tr>
</tbody>
</table>
GLOSSARY

AAPS - Arable Area Payment Scheme
An EC CAP Scheme which is payable on certain arable crops to farmers who, due to reduction of support prices, set aside a prescribed percentage of land in return for compensation.

AO - Area Office.

BSPS - Beef Special Premium Scheme
An EC CAP Scheme that operates on a calendar year basis providing direct support for beef producers. Premium can be claimed twice in the life of a steer but only once for a bull.

CAP – Common Agricultural Policy

CAPM - Common Agricultural Policy Management
The operations side of the Regional Government Department that deals with the CAP Schemes administration.

CASE – Computer Aided Systems Engineering
Software development tools that automate aspects of the development process.

CD[W] - Customer Details [Wales] Form
A regional control mechanism designed to track and maintain customer details and data of all payments made. It is an EC regulatory requirement.

CETD - Cattle Ear Tag Database
A database that holds details of all animal ear tags together with details of their individual subsidy claims in UK to comply with an EC audit requirement that cross border validation is maintained to prevent duplication of payments.

CTS - Cattle Tracking System
A system that registers all the cattle details to enable validation of claims.

DMS - Document Management System
A FileNET service for the New System Project that enables the storing and retrieval of project related documentation.

DO – Divisional Office

EC – European Commission
The section of the European Union that deals with European legislation and its compliance.

EPS - Extensification Premium Scheme
An EC CAP Scheme that provides payment to producers who claim/receive payment under BSPS and/or SCPS that meet certain criteria.

FileNET eProcess
FileNET eProcess is a software package that is used for document management and workflow capability.
GIS – Geographical Information System

ILOG
A set of software tools for developing and debugging business rule applications that allows the rules to be written and modified by non technical users.

ILOG Jrules Business Rules Engine
A mechanism that manages the business rules.

IACS - Integrated Administration and Control System
IACS is the main EC control system that has a direct affect on a number of CAP Schemes. It is an anti-fraud measure designed to prevent abuse of the support arrangements in place for farmers within the EC.

IAD – Iterative Application Development
Commercial system development approach adopted by the Suppliers that they aligned to RAD.

JAD – Joint Application Development
A systems development technique for requirements gathering involving small teams of clients and developers working together in workshop scenarios.

MoSCoW
A systems development technique for prioritising the development of business requirements to meet project deadlines.

MST - Multi-skilled Team
A team of multi-skilled users, employees of NAFW, who are able to resolve the queries that arise in the CAP scheme application claims processing.

Nvivo [QSR NUD*IST Vivo]
A qualitative software package used to store, organise and manage qualitative data electronically.

RAD – Rapid Application Development
An iterative and incremental system development approach that compresses the analysis, design, build and test phases of a development project into short iterative development cycles.

Silo - A small discrete team of people working together.

SAPS - Sheep Annual Premium Scheme
An EC CAP Scheme that supports sheep farmers by providing annual deficiency payments on eligible breeding ewes.

SCPS - Suckler Cow Premium Scheme
An EC CAP Scheme that provides direct support for specialist beef producers of eligible breeding cows.

SSADM – Structured Systems Analysis and Design Methodology
Structured information systems development approach formerly mandated by the Government for development projects.
**Timebox**
Characteristic of RAD that applies a fixed period of time to a specified piece of design/development work.

**TIRs – Test Incident Reports**
Reports resulting from user and system testing detailing problems identified.

**TM - Tir Mynydd**
An EC CAP Scheme aimed at supporting the 'hill farming' of sheep and/or suckler cow producers who farm at least 6 hectares of forage land in less favoured areas.

**UAT – User Acceptance Testing**
Teams of users who test developed parts of the system for usability prior to implementation.

**Workflow**
The computerised automation of the generic business processes standardised across all schemes.
CHAPTER ONE  INTRODUCTION TO THE RESEARCH

1. Introduction

The goal of this research study, through an intensive case study approach, is to examine the issue of utility of a Rapid Application Development approach across a large, complex information system development project.

RAD is an iterative and incremental development approach where the analysis, design, build and test phases of the development lifecycle are compressed into a sequence of short, iterative development cycles. It is these development cycles that are able to accommodate the growing uncertainty and increasingly volatile nature of current development environments. It evolved to address the problems associated with the traditional development approaches such as the Waterfall Model (Martin 1991, Beynon-Davies et al. 1996, 1999, 2000, Elliott 1997, Tudhope et al. 2001).

1.1 RAD – Historical Overview

Rapid Application Development (RAD) as a development methodology has its origins based within the commercial arena. RAD development was driven by the IT industry responding to the need for faster system development than the traditional methodologies could deliver. The shape of RAD has been little influenced by academic input. By the mid 1990s RAD was used as an umbrella term by different vendors applying their own interpretation and approach to encompass a number of methods, techniques and tools. (Beynon-Davies et al. 2001, Osborn 1995, Elliott
1997). This unstructured and extemporised evolution of RAD means that the rationale behind its use is not always clear. It is perceived as a methodology or framework, a method for developers to change their development processes or it is seen as RAD tools to improve development capabilities. As a result individual philosophies and perceptions of its rationale and applicability have led to considerable debate about its utility for large complex Information Systems (IS) development.

Consequently, early RAD was seen as reactive, fragmented and sometimes considered as hype (DSDM 2001). Some refer to it as 1st generation RAD. Thus, in 1994 the DSDM (Dynamic Systems Development Method) Consortium was formed by vendor and user organizations to develop and promote a public domain method - a 2nd generation RAD. Their purpose was to combine the best elements of existing methods with the practical experience of RAD developers into a coherent framework that focused on the principle of fixed timeframes and resources where requirements can be flexible.

1.2 Research Rationale

As a systems development approach, RAD has both critics and supporters whose opinions, in some cases, are fundamental to individual philosophies and perceptions of its rationale and applicability. This has led to considerable debate about its application for large, complex IS development. Even though RAD is becoming an increasingly accepted approach to IS development, there is still considerable debate about its suitability for large complex development environments. Whilst existing literature exposes particular themes of discussion it does little to clarify the position. A prominent area of debate concerns the utility of RAD for large and complex development environments.
Although there is substantial reporting of RAD’s application within the commercial sector there is limited availability of published academic material. For example on the academic front examples of previous research include, Beynon-Davies et al. (1999, 2000) reporting on 7 small/medium studies of RAD development projects, Girling et al. (1998, parts 1 and 2) who inform on a single GUI case study development using RAD, and Jones and King (1998) who review 2 RAD implementations. Other published work concentrates on discussing RAD’s purpose within small to medium development environments. For example Eva (2001) looks at requirements acquisition, whilst Maner (1997), Highsmith (2002) and McConnell (1996) reflect on its speed and prototyping properties with little analysis or assessment across larger contexts.

Even though RAD is becoming an increasingly accepted approach to IS development, much of the existing literature focuses on small to medium sized development projects. There is a tendency to rule out using RAD for large scale projects, thus RAD is typically applied to relatively small low-key projects that are not computationally complex (Martin 1991, Avison and Shah 1997, Beynon-Davies et al. 1998, 2000, 1999, Howard 2002). These views add to the current RAD debate and the literature continues to question RAD application across such development environments, and posits that the lack of academic research in this area identifies a need for further evaluation of RAD-type projects (Beynon-Davies et al. 1996, Boehm 1999, Elliott 1997, Howard 2002, Osborn 1995).

Contrary to the published beliefs, a RAD-type approach is being adopted for a large, complex information system that is currently being implemented within UK Regional Government that is being used as a case study by the researcher. This
The case study presents an interesting and atypical opportunity to examine the utility of RAD within such a complex development environment.

It is the current debate surrounding RAD that interests and motivates the researcher, and the case study opportunity enables the researcher to respond to some of the views and opinions expressed in the literature reviewed and contribute to the limited academic reporting.

1.3 Aims and Objectives of the Research

The goal of this research study is to identify the issues that impact and influence the application of a RAD-type development approach within a large and complex development arena. The research aim and objectives are set out below:

Research Aim

To analyse the utility of using a Rapid Application Development (RAD) for a large, complex information system.

Research Objectives

- to determine the current status of RAD through a critical literature review.
- to examine the application of a RAD approach on a large and complex IS project.
- to identify the core components of a RAD-type development approach, and establish their presence within the case study project.
- to determine the issues that impacted upon, and influenced the utility of the development approach.
to present a model of critical success factors for large and complex developments adopting a RAD-type development approach.

Four research questions have been created in order to achieve these goals; these are set out below:

Research Question 1: Which issues are considered to have the most effect upon a RAD-type development?

Research Question 2: What are the core features of a RAD-type development approach, and were they applied to the case study project?

Research Question 3: What are the major influencing factors that affected the development process of the research case study project?

Research Question 4: Were the problems experienced attributable to the RAD-type development approach adopted, or to other aspects of research case study?

1.4 Introduction to the Case Study

The case study concerns the development of a New IT System that is currently being implemented within UK Regional Government. The system is aimed at improving the current administration of the EC’s Common Agricultural Policy (CAP) grants and subsidies schemes across the region. Improvements were deemed necessary due to a history of late payments, poor customer satisfaction and an increasing inability to meet the EC’s changing requirements. Due to the nature of the EU’s agricultural policy, schemes are frequently changed with new schemes being drafted as required. Every scheme must conform to the EC legislation and regulatory control mechanisms that introduce an evolving and dynamic nature to the business environment.
The case study project was initially planned for 2 years but overran into a 3rd year. The size and complexity of the New IT System are reflected by an initial cost estimate in excess of £10m; a core project team of 50+; and the extent of the customer base that is measured in terms of 100,000s of annual grant and subsidy applications that are spread across the UK region. The new system moves away from the former individual scheme administration procedures towards a Generic Process that integrates the core processes across the common activities of the separate CAP schemes.

Development was outsourced to a commercial company but the project environment remained within a central location where both clients and developers were co-located on the same site for the duration of the project. The developers adopted their own in-house commercial Iterative Application Development (IAD) approach to promote a controlled, structured but flexible development method aimed at providing incremental delivery that can be aligned to RAD. They believed that this method offered all the main benefits of a RAD type approach and that it was suited to the uncertainty of, and continually changing business requirements. IAD, like RAD, involves prototyping and iterative delivery but without the problems of lack of rigour, creeping feature scope and overrun that are thought to be associated with a RAD approach.

1.5 Introduction to the Research Approach

An interpretive research method was adopted through an intensive case study approach that utilised ethnography as a qualitative data gathering method. Literature recognises that case study utilization is especially suited to interpretive research of information systems (Yin 1993, Myers 1997, Orlikowski and Baroudi 1991). This approach was selected because the researcher wanted to gain a
broad understanding of the organizational environment of the IS and the wider external context within which the information system was related (Walsham 1997, Gill and Johnson 1991). There are many instances of this research style being adopted within IS research by authors such as Beynon-Davies et al. (2000), Myers (1999) and Walsham (1997). The intent of interpretative research is not to generalise but to understand the deeper structure of a phenomenon such that it can be used to inform other similar settings (Orlikowski and Baroudi 1991). In other words, to get an increased understanding of the phenomenon being investigated within its cultural and contextual situations inside their natural setting. This enabled the researcher to more fully understand the nature of the problem than a quantitative analysis would allow. The case study approach provided a unique opportunity to investigate the research questions within a ‘live’ project environment, involving real-life experiences and commentary.

Ethnography was selected as a style of interpretative qualitative research because it promotes intensive data collection that allows a broad and rich interpretation to be gathered (Myers 1999). Its suitability is reflected by its association with IS research (Loftland and Loftland 1984, Strauss and Corbin 1990, Beynon-Davies 1997).

Analysis was drawn from both primary research and secondary research practices. The primary research concerned direct observation, indirect observation, informal semi-structured interviews, shadowing of key informants and spontaneous conversations, while the secondary research reflects an analysis of published literature from both academic and practitioner perspectives and examination of project documents, discourse and artefacts.
1.6 Thesis Contributions

The significance to the Information System field is detailed through the thesis contributions, these are set out below:

The thesis contributions

- A systematic and comprehensive literature review that addresses both academic and practitioner environments which serves as a reference for other researchers interested in Rapid Application Development.
- The study provides a rich, ethnographic case study account of a large complex RAD development project to address the recognised lack of such material within the IS field, and responds to some of the opinions expressed therein.
- There are numerous examples of small and medium RAD projects, mainly driven by the commercial sector, this thesis presents particular benefits as it provides insight into the actual practice of the RAD-type approach within a large and complex IS development with some real life context, experiences and commentary from individuals directly involved in case study.
- It extends and supplements previous research by reporting on the most critical processes affecting a RAD-type approach within this type of development milieux.

Thus it can be said that this research study has examined, extended and made original contributions to the IS field through the investigation of a ‘live’ project environment of a RAD-type project that are relevant to future IS development projects considering adopting such an approach.
1.7 Organization of the Thesis

This doctoral thesis is organized in nine chapters. The initial three chapters deal with general concepts — thesis introduction, the research aim, objectives and research questions to be answered, the research methodology and literature review. The following chapters present the case study and the analysis that is presented in two parts. Finally the conclusions chapter addresses the four research questions, presents a model of critical success factors, explains the constraints and limitations of the research study and the issue of researcher’s influence on the study, and puts forward directions for further research.

Chapter 2

Chapter 2 describes the research methodology approach adopted. The researcher chose to present this chapter before the literature review that is contrary to the more traditional thesis structure, because the literature review was a complex and extended matter that is spread over two chapters. In this way the literature review is aligned closer to those issues analysed and evaluated. The purpose of this chapter is to present the thesis' conceptual framework, and to expound the research rationale and strategy and the associated research methods and techniques employed.

Chapter 3

This chapter sets out the literature review for the research study. It aims to present an understanding of the current status of RAD, and its position within the IS field and provide a foundation for an understanding of the current status of RAD, such that the reader is able to fully appreciate RAD’s emergence. It presents a brief history of the development of Information Systems, a synopsis of software development environments and an overview of advances in development methods. From this analysis a genesis of RAD is presented and a table of critical
RAD factors created from the opinions and views expressed in the literature within a current context of its application as a development approach.

**Chapter 4**

Chapter 4 represents a further literature review that extends the initial RAD centred review to deeper analyse the issues impacting upon the development approach as they emerged from the ethnographic case study. Issues concerning user involvement, requirement elicitation, decision-making, user expectation, and communications all required further examination within the wider literature rather than just from a pure RAD aspect.

**Chapter 5**

Chapter 5 describes the context of the case study scenario and is divided into four sections. The first section sets out the background, historical and organizational perspectives, together with the rationale behind the new system and the procurement process involved. The second section presents a high-level view of the New IT System. The third section gives an overview of the system and technical architecture that supports the New IT System. Finally, section four describes a high-level view of the development approach that was applied to the case study project and the extent to which the Business Vision was achieved.

**Chapter 6**

This chapter defines and characterises the main research themes that have been associated with cultural influences, and the development factors that impacted on the research case study project. It looks at two key aspects of the cultural influences involved, those of organizational structure and organizational culture with specific reference to the nature of the inherent bureaucratic culture that is associated with such public sector bodies. This chapter further discusses the relationships that exists between the research themes and sets out the general
structure and framework for the data analysis of the case study materials gathered.

**Chapters 7 and 8**

Chapter 7 and 8 present the analysis of the data gathered. Due to the extent of the analysis achieved through the ethnographic research approach it has been divided into two chapters that represent the natural divide of the data analysed. Chapter 7 analyses issues concerned with user involvement, user expectation and communications. Chapter 8 represents the analysis of the problems experienced with requirements elicitation, decision-making and testing.

**Chapter 9**

Sets out the conclusions drawn from the analysis and reflects on the views and opinions expressed in the literature review with the aim of questioning or validating the published belief that RAD is not suited to large and complex development environment. Each research question is revisited and discussed. The contributions and key findings of the study, along with the limitations of the study are also presented. Finally the validity of this doctoral research is evaluated and directions for further research are put forward.
2. Introduction

The purpose of this chapter is to set out the research philosophy that guides the research, its design and methodological approach. An overview of the case study context and of the research approach are presented to provide clarity and understanding for the reader. This chapter continues with a justification of the philosophical perspective and then explores in more detail the rationale of the research design, methods and techniques and their suitability to this research study.

It introduces the research philosophy of phenomenology i.e. qualitative research, sets out the rationale behind the choice of ethnography as a style of interpretative research and examines the case study approach. The suitability of the research design for this research study is further justified and aligned to previous research studies.

The data collection and research techniques for both primary and secondary research data gathering are set out and examined in detail taking into account recognised concerns expressed in the literature. The interview population is described and the interview process is clarified through its advantages and disadvantages. Data gathering techniques and analysis processes are explained, and the rationale behind the use of NVivo qualitative software for data organization is presented.
The reliability of the research and the application of ecological validation are examined together with the researcher’s creditability, and ethical issues associated with the project are identified. The chapter concludes by looking at the strengths and weaknesses of the research strategy chosen, and finally the research schedule is outlined and explained.

2.1 Case Study Context and Research Approach.

Set out below is the context of the case study, and an overview of the research approach adopted.

2.1.1 Overview of the Case Study Context

In 1999 under the UK Government’s Devolution legislation a UK Regional Government department took on the devolved functions formally carried out by the Welsh Office. It became responsible for managing the expenditure of EC grants and subsidies to customers through a number of Common Agricultural Policy (CAP) schemes across the region.

The case study concerns the development of the New IT System aimed at improving the current administration of the EC’s Common Agricultural Policy (CAP) grants and subsidies. The system moves away from the former individual scheme administration procedures towards a Generic Process Model (figure 1.1) that integrates the core processes of the common activities of the separate schemes.

![Figure 1.1 Generic Process](Source: Adapted from Project Ref DT.01.001.01)
Each scheme incorporates the related EC rules and conditions that apply, however schemes do not exist independently of each other, but acquiesce to a 'network' of complex interdependent relationships that must conform to EC legislation and regulatory control mechanisms that undergo continual change. Formerly, schemes were the responsibility of individual Scheme and Process Managers who attended to the business needs and administration respectively.

The Project is described as large. Its size and complexity are reflected by an initial cost estimate of £10m, a projected timeframe of 2-3 years, a core project team of 50+, and the extent of the customer base that is measured in terms of 100,000s of grant and subsidy applications across the region.

Development was outsourced to a commercial company but the project environment remained within a central location where both clients and developers were co-located on the same site for the duration of the project. The developers adopted their own in-house commercial Iterative Application Development (IAD) approach to promote a controlled, structured but flexible development method aimed at providing incremental delivery that can be aligned to RAD. They believed that this method offered all the main benefits of a RAD type approach and that it was suited to the uncertainty of, and the continually changing business requirements.

2.1.2 Overview of the Research Approach

The researcher adopted an interpretive stance as advocated by Walsham (1997) aimed at producing an understanding of both the context of the IS and the process in which the IS influences and is influenced by its context. Ethnography was selected as a style of interpretative qualitative research for intensive data
collection that allowed a rich and deep interpretation (Myers 1999, Orlikowski and Baroudi 1991). This enabled the researcher to get an increased understanding of the phenomenon being investigated within its cultural and contextual situations inside their natural setting, and to more fully understand the nature of the problem than a quantitative analysis would have allowed. The case study approach provided a unique opportunity to investigate the research issues within a ‘live’ project environment.

Analysis was drawn from both primary research and secondary research practices. The primary research concerned direct observation, indirect observation, informal semi-structured interviews, shadowing of key informants and spontaneous conversations. Secondary research involved an in-depth analysis of published literature from both academic and practitioner perspectives and examination of project documents, discourse and artefacts. More specifically, secondary research reflected three levels. Firstly, an in-depth exploration and analysis of existing RAD literature from both academic and practitioner perspectives to present a foundation for an understanding of the current situation in the field of Rapid Application Development. Secondly examination of development issues within the wider IS domain. Thirdly, an examination of existing project documents, discourse and artefacts. This allowed cross checking and was aimed at yielding a stronger substantiation of analysis and conclusions drawn.

A Case Study database was created using QSR NUD*IST Vivo (NVivo), a qualitative software product to store and analyse the range of qualitative data collected (Myers 1999, Yin 2003). Initial data analysis was driven by the data rather than the researcher and concerned ‘open coding’ that involved analysis of the content where data were analysed and categorised into themes. Further
investigation necessitated establishing how these categories might inter-relate and link into sub-categories that reflect the associated dimensions and properties where data are further organized by reoccurring theme (Orlikowski 1993). This is commonly known as axial coding and was used to uncover the relationships within the categories.

2.2 Research Philosophy - Phenomenology

Since the 1980s there has been a shift in information systems research away from a predominantly positivist model towards a more phenomenological perspective involving qualitative methods (Myers 1999). Qualitative research encompasses a broad range of research approaches. There is no one recognised qualitative approach; each piece of research is dependent on the variables involved. These include the research goals, the researcher's ontological standing, the nature of the knowledge and how it is acquired and so on (Ritchie and Lewis 2004). It is therefore important that the reader understands how this research study is situated within the broader field of qualitative research. Qualitative research is generally associated with interpretivism that deals with social investigation, rather than positivism that subsumes the laws and regularities more suited to quantitative research and statistical outcomes.

However, it is the application of a philosophical framework that determines the suitability of both the researcher's views of reality and a particular research approach to a research study. Myers (1999) believes that it is important also to consider the underlying epistemology which guides the research. This refers to the assumptions about the nature of the social world, the nature of the knowledge gained and how it is acquired and how to determine whether or not something is true (Ritchie and Lewis 2004, Myers 1997). It provides a foundation that enables
both evaluation of knowledge by specifying what is acceptable, whilst being able to
distinguish it from incorrect supposition (Johnson and Duberley 2000). The
adoption of a research method itself is not sufficient to gain that understanding,
therefore the researcher must adopt a coherent philosophical approach for such
an understanding to be achieved.

It is important that the research method chosen is appropriate vis-à-vis the
research aims and objectives, the nature of the problem to be explored, and the
constraints imposed in terms of the time and resources available to conduct the
research (Myers 1999). Consequently it was necessary for the researcher to
understand the nature of phenomenology. However, it is not necessary to return to
the founders of philosophy considering Plato’s world of ideas or Aristotle’s world of
the senses via Descartes and the early 18th century pragmatists (Hume and Kant
- cited in Nicholas 1996) but rather, as Myers (1999) suggests to adopt Husserl’s
20th century perspective of focussing on the human experiences of the world on
the individual experiences, rather than on the world itself (cited in Nicholas 1996).

Myers puts forward Husserl’s observation that consciousness is always directed
toward something. He argued that consciousness contains ideal, unchanging
structures called meanings, which determine what object the mind is directed
toward at any given time. However early criticisms of phenomenology were that it
was essentially a solipsistic method, in other words confined to the view that ‘self’
is all that exists or can be known.

Phenomenology relates to understanding the meaning that lies within social
interaction and in individual actions in the world, because experiences are based
upon, and influenced by social meanings and not upon random individual
reflections (Loftland and Loftland 1984, Strauss 1973, Myers 1999). It is necessary to explore why individuals act the way they do, and what common understandings and behaviour result. This is compared to a positivist perspective that relies on the law of large numbers to provide generalizable conclusions through statistical analysis.

The argument here is that for such generalizations to be made there is a need to understand the problems being investigated at the level of significance to, and actions of, the individuals concerned. This activity is more suited to phenomenological methods, such that conclusions drawn are more likely to be based on a deep awareness of the details of individual realities and how they are affected. As a consequence ethnography has been chosen as the research design because it represents one of the most in-depth research methods providing rich insights into the human, social and organizational aspects of a situation (Myers 1999, Ritchie and Lewis 2004), and thus falls well within the phenomenological remit of the research philosophy.

2.3 Research Design and Methodology

The researcher chose a qualitative research approach for the investigation into the case study of the new IS development project. In terms of methods, perspectives and strategies qualitative research is an umbrella term that encompasses a variety of approaches, and ethnography was selected as the most appropriate.

Ethnography is a style of exploratory qualitative research that has become associated with the development of information systems (Gill and Johnson 1991, Beynon-Davies 1997). This coupled with the duration of the project, lends itself to a longitudinal study involving ethnographic methods that are perceived as the best
use of the researcher’s time and the constraints of the project. Additionally, the accessibility and availability to the data combined with the high-levels of participant co-operation achieved also justified this approach. Moreover as the research project itself in many senses is of an exploratory nature it is felt that an ethnographic approach is further emphasized.

The research strategy adopted not only takes account of the philosophical issues but also the structural issues involved. This refers to the way the research is sensitive to the understandings of the subjects being researched and also that interpretative consideration is given to the social and organizational contexts in which the actions take place. This facilitates broader and richer descriptions and applies sensitivity to the social meanings they construct and reconstruct and thus reflects the phenomenological approach taken.

This research lies within the conventional interpretation of ethnography with a commitment to observation activities combined with informal semi-structured interviews that involved multiple strategies of data collection and analysis. Conventional ethnography seeks to describe the situation as it is without criticism, compared to a critical ethnography that is orientated towards exposing oppressive practices that do not reflect the natural order and where participants are seen as complex and ambiguous (Johnson and Duberley 2000).

Qualitative research has been identified as providing a richness and depth of data on the concerns of process and/or practice over time compared to a snapshot view of a quantitative approach. How a RAD-type approach was used in practice during the development project is a key focus of this research and therefore an ethnographic research approach was considered appropriate.
Additionally the position made for adopting a qualitative design and in particular ethnography is strengthened by the suitability of this approach to achieving the research questions. This pertains to research question 2, but more specifically for research questions 3, and 4. For example research question 2 necessitated the identification of the core features of a RAD-type development approach that was accomplished through the systematic and comprehensive review of RAD literature (as was research question 1). This question also required the researcher to establish the application of the core features to the case study project. Observational evidence provided rich data that allowed validation of findings through interviews that further verified analysis. Research questions 3 and 4 were concerned with identifying the influencing factors that affected the development process of the project in practice, and whether they could be attributed to the development approach or to other aspects of the research case.

Ethnography enabled the researcher to gather a depth of data that increased the understanding behind the issues as they emerged that could not have been achieved through interviewing alone. It further facilitated acknowledgement of the social and cultural influences that could have been lost or missed and gave greater significance to the conclusions drawn.

2.3.1 A Case Study Approach

Case studies are strongly associated with qualitative research. The utilization of a case study as the mechanism for this research project fell comfortably within the proposed ethnographic stance and interpretive approach of the research design. Yin (1993) recognises that case study utilization is especially suited to exploratory research design because it encompasses similar qualitative research techniques. Moreover it is particularly suited to IS research because the object of the discipline
is the study of the impact of information systems on an organization and its environments as is the case in this research project (Yin 1994). Thus it allows the researcher to investigate the phenomenon in its natural context, drawing on multiple perspectives, that is critical to the understanding of the key themes, concepts and issues involved (Ritchie and Lewis 2004). Therefore exploring behaviour and social actions to determine how people interpret and react to their environment enables the researcher to analyse and identify emerging categories of social action and the relationships between them. This is associated with grounded theory (Glaser and Strauss 1967) and concerns the discovery of theory from data of the phenomenon under study that has been systematically obtained and analysed.

Additionally case study research facilitates a more detailed view of the issues under investigation to be developed than would be available through a quantitative approach. This view is also supported by Myers (1997), who states that the case study method is particularly well suited to IS research, and Orlikowski and Baroudi (1991) describe it as the most common qualitative method used in IS research. Yin (1994) defines the scope of a case study as one that investigates an existing phenomenon within its real life context and this is the situation with this research case study.

2.3.2 Suitability of the Approach to IS research

There has been much debate about the types of research conducted within the field of IS (Benbasat and Zmud 1999, Davenport and Markus 1999, Lee 1999). An overriding concern was to ensure that the research approach employed should be relevant and suitable to IS field such that it would satisfy the research aims and
objectives as well as be sufficiently rigorous to enable valid conclusions to be drawn.

As previously stated, the phenomenological, philosophical and ethnographic style of interpretative qualitative research was selected because it promotes the intensive data gathering techniques that allow a broad and rich interpretation to be gathered (Myers 1999), that enables rigorous triangulation of the various data collection methods employed. An assured contribution that will add to, and extend the identified debate with the IS development arena is thus enabled.

The suitability of this approach is reflected by its association with IS research (Loftland and Loftland 1984, Strauss and Corbin 1990, Beynon-Davies 1997). There are many instances of this research style being adopted within the IS field by authors such as Beynon-Davies et al. (2000), Myers (1999), Jones and King (1998) and Walsham (1997). The intent of interpretative research is not to generalise but to understand the deeper structure of a phenomenon such that it can be used to inform other similar settings (Orlikowski and Baroudi 1991). In other words, the purpose is to get an increased understanding of the phenomenon being investigated within its cultural and contextual situations inside their natural setting. The suitability is further reflected in the argument that the social world cannot be understood in terms of simple causal relationships or assumptions of universal laws because experiences are based upon or infused by social meanings (Loftland and Loftland 1984, Strauss and Corbin 1990, Hammersley and Atkinson 2000). This enables the researcher to more fully understand the nature of the problem than a quantitative approach would have allowed.
Additionally, the literature recognises that case study utilization is especially suited to interpretive research of information systems because, as mentioned above, it encompasses similar qualitative research techniques, and consequently it falls comfortably within the proposed ethnographic stance of this research design (Yin 1993, Myers 1997, Orlikowski and Baroudi 1991).

2.4 Data Collection and Research Techniques

Data collection concerned both primary and secondary resources. Primary refers to those data that are unpublished and which the researcher has gathered from people within the case study organization who are directly involved with the IS development of the IS Project. A variety of fieldwork methods were used to collect primary data. These included both direct observation and indirect observation, informal semi-structured interviews, shadowing of key informants, informal discussions and spontaneous conversations that were conducted within the case study scenario of the IS Project. Each of these methods is described in more detail below.

Secondary research was conducted on two levels. Firstly, an in-depth exploration and analysis of existing literature from both academic and practitioner perspectives to present a foundation for an understanding of the current situation in the field of Rapid Application Development. For greater insight and to place RAD in context, the literary review was extended to include a brief history of information systems, the evolution of development environments, and of associated development methodologies. The literature review was purposeful in not only determining areas of conflict and contradiction within the field of systems development, but also to research the current debate surrounding the use of RAD and draw out the specific issues pertinent to this research study.
The researcher sourced books, journals and literary papers through the use of the university library and other libraries using the inter-library loans facility and On-line Public Access Catalogue (OPAC) searching that included subject related on-line databases. Although use of the Internet as an electronic source of data collection was employed it was restricted to verified sources of information.

Secondly, due to the late entry of the researcher into the IS Project environment an initial focus concentrated on the examination of existing project documentation for the researcher to form a basic understanding of the organizational and social contexts involved and to establish the status of the current IS Project. The social and organizational contexts were identified as critical to an understanding of the situation because they defined the social policy reasons for the new system development. The researcher acknowledged the importance of existing documentation as a secondary data source because they would not only provide the necessary background data but also present a basic source of information about project decisions, activities and processes prompting further lines of enquiry.

However it was also recognised that a positive spin 'before substance' premise occurred in the some of the project documentation and the recording of events. Although operationally this approach is acceptable to the organization, the researcher realised the importance of taking this into account when examining the official project documentation. Accordingly some allowance was made for the intended audience and the possible motives the author may have had in presenting a positive outcome. Furthermore data gathered from this source was subjected to rigorous triangulation within the data analysis process.
2.4.1 Direct Observation and Indirect Observation

Essentially exploratory research involving an ethnographic approach necessitates a sustained period of fieldwork by the researcher in order to get close to the organization and the participants being studied (Alvesson and Deetz, 2000). Hammersley and Atkinson (2000) define it broadly as a study of an exploratory nature working with unstructured data that is case oriented and interested in meanings.

The key characteristic of ethnography is that of observer participation, analysing behaviour by observing events as they occur, or do not occur, in practice in their natural context (Silverman 1985 cited in Alvesson and Deetz 2000). Observation is a central tenet of ethnography and the researcher believed that the use of observations in the natural setting would provide a wide and rich perspective of the issues under investigation. However, the researcher recognised that due to the time constraints involved and the high degree of personal involvement that is required it would not be possible for observations to take place on a continuous basis. Consequently, a significant emphasis was placed on indirect observation and other techniques such as interviews, spontaneous conversations / discussions, and review of documents as substitute data collection methods. Indirect observation occurred where the researcher, unable to attend some events, sought feedback and information from participant informants either orally or in writing (Gill and Johnson 1991).

A major concern of using observation as a data gathering technique is the effect the observation has on the participants being observed. This is recognised as the
Hawthorne effect\(^1\) (Preece \textit{et al.} 1994, Silverman 2001) that questions whether participants who know they are being observed change their behaviour. Literature suggests that there may be some justification for covert observation to counteract this effect (Patton 1990, Phillips and Pugh 1998). However the research design for this study prescribed an initial period of intensive non-participatory observation over several months that enabled the researcher to become established within the IS Project environment and build up a relationship of trust with the participants. Consequently those involved became willing participants giving full co-operation such that covert observation was not necessary.

Observational focus concentrated on data collection from observing people working in their day-to-day team environments, attending both formal and informal meetings reflecting different organizational levels, attending project presentations and briefings and generally watching and listening as the project progressed. The observations led to a large volume of extensive field notes. Myers (1999) suggests keeping a careful record of field notes to avoid problems later on with tracking what occurred, when, why and how. Consequently, these were initially recorded regularly in a Project Diary, as suggested by Silverman (2004), together with a variety of supporting documentation such as would be given out in meetings. However as this was time consuming the researcher preferred to record observation notes, feelings, impressions, ideas and the questions that emerged from observations electronically at the end of each day. This data had the added advantage of being able to be directly imported into the qualitative software program QSR NUD\(^*\)IST Vivo (NVivo), along with electronic versions of the supporting documents where it was organized, managed and subsequently coded.

\(^1\) Hawthorne Studies 1920s/30s - Lead by Elton Mayo at Hawthorne Plant of Western Electric Co.
Chicago.
for analysis. This is more fully explored in Section 2.5. Furthermore it facilitated the ability to audit field notes that enabled the researcher to track the data with associated thoughts and ideas throughout the research study (Silverman 2004).

2.4.2 Informal Semi-structured Interviews

Semi-structured informal interviews were chosen as a specific method of data elicitation to complement the observation activities. Qualitative interviews enable rich in-depth accounts of interviewee experiences, knowledge, ideas and expressions, and thus an understanding of what occurred (Alvesson and Deetz 2000). Additionally they allow the researcher to explore in broader context the ideas that may not have been previously acknowledged (Myers 1997).

Traditionally qualitative interviewing adopts a question and answer technique where the power lies with the interviewer who controls the data collected. De Laine (2000) suggests moving away from this stimulus response model with its inequitable power between interviewer and interviewee towards a more interviewer guided style of phenomenological interviewing that empowers the interviewee. However Silverman (2001) points out that if the interviewer remains too passive then this can itself create a constraint on the interviewee. Therefore the researcher conducted informal semi-structured interviews that guided but not controlled the data collection. An advantage is that whilst both parties are free to explore within a particular subject area, it allows the researcher to make best use of time limitations. Additionally this approach also facilitated a systematic framework across a number of interviews to ensure that core issues were addressed.
2.4.2.1 Interview Population

The project environment combined two different cultures that were co-located on site for the duration of the IS project. Firstly the Client culture of a bureaucratic government department, and secondly the commercial culture of the outsourced Supplier’s developers. The interview strategy explored above was applied across both settings. Interviewees represented project members who were actively involved in the IS project. In total 126 interviews, over 90% of the core population, were accomplished across the core project team, transcribed and used for analysis. The number of interviews per person was dependent upon their status within the project, their potential contribution, their availability and willingness to participate. For example project and business managers were interviewed between 3 – 6 times compared to the more peripheral project participants who were accessed once. The researcher aimed through repeat interviewing over time not only to gather sufficient data but also to validate their consistency facilitating the data analysis process. Nevertheless the researcher recognised that the interview situation itself becomes more difficult when interviewing relatively powerful people who are used to being in charge (Hammersley and Atkinson 2000). Through the use of the interviewer guided style adopted by the researcher that empowers the interviewee this was not an issue.

However some problems were experienced, firstly because of the late entry of the researcher into the project arena a few of the development team had been ‘rolled off’ the project. Consequently these individuals were contacted and interviews arranged at their convenience outside the project environment, some were conducted by email. Of those contacted only two felt unable to contribute due to pressures of work and their distance from the project.
A further problem encountered involved the practicality of conducting the interviews due to the pressure of work the employees were experiencing. However this problem was overcome by introducing flexibility into the interview sessions whereby the researcher would be available on site for a whole day so interviewees could attend when convenient within the timeframe allotted – this was very successful. There was no evidence of any organizational influence exercised over the interviewees, or of their delivery, and all interviewees gave informed consent.

Interviews were conducted on a one-to-one basis, sessions were scheduled and conducted on site, were generally of an hour’s duration, but did not exceed two hours, and all questions posed were open ended in order to gather data that was both rich and deep. Due to the levels of project activity and pressure to maintain the Business as Usual activities there was no hierarchical order to their occurrence, interview schedules were organized around interviewees’ availability. This structure was the only practical arrangement. Where feasible, interviews were held as close to naturally occurring events as possible i.e. where development was delayed the researcher made efforts to communicate with those involved as soon as possible. In total the range of interviews conducted provided data that were both in-depth and rich, and which also enabled validation through triangulation techniques.

A core set of questions was defined for all participants in order to establish a pool of general data from which some common conclusions could be drawn. Thereafter interviewees were asked about specific areas to cover the central research issues, and subject specific questions relevant to each participants project role. Questions were designed not only to elicit qualitative data to answer the research questions, but also to correspond with a range of initial values that facilitated the
early organization of qualitative software coding. More specifically, they were questions about interviewees' knowledge in order to extract factual information, and about their experience and behaviour to describe actions and activities that had not been observed that assisted triangulation as data could be verified.

All interviews were audiotape recorded with the informed consent of the interviewees. Each tape was fully transcribed verbatim by the researcher within two days as recommended by Myers (2000) and Patton (1990). Respondent validation was applied whereby a transcript of each tape was returned to the relevant interviewee for verification and feedback. Apart from validation purposes the researcher determined that taping interviews had several clear advantages. It captured the exact data in an unbiased format and provided a record that could be revisited for clarification at a later date. It stored the style, tone, inclination and sequence of the interview. It enabled the researcher to concentrate on the interview and not on note taking.

Although transcribing tapes took a considerable amount of time, it is acknowledged that this task enabled the researcher to gain valuable insight upon a second hearing during the transcription process. In the personal interview scenario a lot of time is spent on interacting with the interviewee. However through transcribing without this face-to-face contact the researcher was able to hear and analyse some tonal or ambiguous implications that were later investigated.

Lastly, the researcher acknowledges the potential of bias where individuals who feel they have been adversely treated within their working role can often present a negative view or opinion, in the same way that a positive prejudice can apply
(Alvesson and Deetz, 2000). Consequently triangulation played an important role in dealing with this problem.

2.4.3 Shadowing of Key People

The researcher's late entry into the project combined with the size and complexity of the IS Project did not facilitate easy integration into the project environment. Consequently shadowing of key people was undertaken during the initial stage of the research study in order for the researcher to fully comprehend the status of the IS Project and the substance of the system concerned. This included the project and business managers, and team leaders across the development arena.

The exercise was only partially successful in that the researcher was not kept fully informed of those people's commitments and was frequently unable to locate and track their movements. Additionally within the development setting some shadowing activities were not particularly informative. This was due to the nature of the development process that consisted mainly of working with a level of system detail that was not the remit of this piece of research. However, a valuable benefit was realised in the identification of these people as key informants.

Patton (1990) recognises one of the main stays of ethnographic fieldwork as that of using key informants. The key informants, as identified above, possessed particular knowledge and insight and not only kept the researcher informed but also became a source of explanation of what had been observed but not understood. However the researcher acknowledges that these contributions present perceptions and not necessarily truths, their data was therefore subjected to validation activities accordingly.
2.4.4 Spontaneous Conversations

A final source of primary data was that of spontaneous discussions and conversations which occurred during informal talks in the natural flow of interaction between the researcher and project members. Both Patton (1990) and Preece et al. (1998) recognise this as being as important as formal communication and that it represents an important aspect of the work environment. Comments and notes were recorded in the researcher's diary and subjected, along with all the other data collected, to the validation and verification processes.

2.5 Data Organization and Analysis Techniques

The research methods as outlined above produced a range of text based and recorded data. Both stand alone and networked PC stations in different locations appropriate to the research being conducted, were used to store and retrieve the data. A back-up strategy employed individual hard drives, floppy disks and memory sticks as back-up copies. Recorded audio information was maintained on traditional cassette audio tapes and were archived after transcription for reference purposes.

QSR NUD*IST Vivo (NVivo), a qualitative software product, was used to organize and manage the data collected within a qualitative context. NVivo is recognised as particularly suited for researchers working with non-numerical data such as that produced from interviews, documents and tapes.

2.5.1 Data Capture and Data Analysis

NVivo was selected to collate the range of qualitative data collected. Its suitability lies in its ability to deal with large volumes of non-numerical data such as
interviews, documents and tape transcripts necessary for the management and analysis of data generated during this research study. Importantly it is in-house to the university where it is considered to be a stable and tried and tested system that is backed up with on site experience and technical assistance.

Moreover it offers different ways of achieving discovery and clarification using integrated coding with qualitative linking of data whilst facilitating shaping and modelling. However as Myers (1999) emphasizes qualitative analysis software packages are useful for organizing the indexing, coding and categorising of data, but it is the researcher who must apply a development strategy to formulate the analysis. Its rationale can be extended to the mapping of the software facilities and characteristics against the research approach, methods used and type of data that the research is generating.

An early criticism of such software management methods suggests that they can be influenced by grounded theory, and as such may push analysis in one direction with the result that some aspects of analysis might be a product of the technology used. However the sophistication of NVivo software means that it is not connected to any one analytical approach (Richards et al. 2002). Additionally software management tools have a reputation for providing speed and flexibility, and the researcher was aware that they also have the potential to restrict and constrain the handling of different data types through their coding, and linking processes. For example a hierarchical organization of data can restrict a flexible and creative analysis of non/semi-structured interviews. However NVivo is designed to work in a networked and integrated way. It supports text, word documents, spreadsheets, multimedia such as video clips and sound bites (audio or video files), whilst at the same time allowing the coding of free independent 'nodes'/categories that can
represent emergent themes or concepts from initial data collection. There is no need to define or specify categories at the onset of the research, although due to the anticipated volume of data the researcher chose to organize the data early. The emphasis is on remaining close to the data. NVivo enables the researcher to return to, retrieve and maintain the contextual integrity of the data, whilst facilitating editing activities without disturbing or invalidating existing coding and linking. Thus it offers both loose and creative structures that are unaffected by editing enabling the researcher to apply perceptions, interpretations and theory development appropriate to the research study.

However, the researcher reassures the reader that it is not possible to eliminate the role of the human in the analytical process, consequently NVivo was utilised as an aid, not as a means of replacing the analysis process. It cannot replace the analytic activities required for the research. Therefore it was not used exclusively, and manual methods were employed in the examination of thematic ideas to establish a depth of understanding of the data.

2.5.2 Application of NVivo to the Research Case Study.

A case study database was created to store and analyse the range of qualitative data collected (Myers 1999, Yin 2003). NVivo was additionally used for the data generated during the literature review activities. NVivo facilitated the systematic building of arguments and consensus that involved reading and reflecting, interacting with the research data to identify key themes, coding the themes, linking similar/contrasting ideas and contradictions of the views expressed, identifying quotes for use, and finally building/creating some original and independent analysis and recording the researcher’s developing thoughts with potential links to supporting evidence.
Initial coding was categorised into values/themes that are linked to the research aims and objectives and organized into a conceptual structure that was methodically coded. This represented the main question topics. For example ‘user involvement’ was represented by an initial ‘free node’ that represented and stored all interviewee responses to that particular question. This activity was followed by rigorous use of appropriate structured approaches to establish how these categories might inter-relate and link into sub-categories that reflected possible associated dimensions and properties. In this way initial data analysis was driven by the data rather than the researcher and concerned ‘open coding’ that involved analysis of the content where data were analysed and categorised into themes. Further investigation necessitated establishing how these categories might inter-relate and link into sub-categories that reflected the associated complexities and properties of the project, where data are further organized by reoccurring theme (Orlikowski 1993). This is commonly known as axial coding and was used to uncover the relationships within the categories. Strauss and Corbin (1990) substantiate the importance of understanding the relationship between structure and process, which they believe are inextricably linked, in order to comprehend the phenomenon being studied. Thus NVivo was used to create a case study database that stored and organized the range of qualitative data collected (Myers 1999, Yin 2003).

In this way theories are not generated prior to data collection and therefore any relevant variables within the data were not predeterined. The researcher was aware of the importance of analysing what emerges from the data itself out of the process of inductive reasoning compared to a statistical inference of a positivist approach (Walsham 1997). For example initial coding was then further coded into simple ‘tree nodes’ representing the main issues emerging from responses to a
specific question topic. In the example used above, the initial 'free node' of user involvement, the emerging issues were (1) identification of the right users, (2) user availability, (3) user allocation. At each level new codes and new links were established across the common fields of data, for example ‘user involvement’ was linked with the node collating business management attitudes.

2.6 Reliability and Ecological Validation

This section informs on two issues prominent within ethnographic research, firstly entry gained to the project and the project community, and secondly the role of the researcher within the research project. Additionally it looks at the issues of the reliability and the validity of this piece of research, and explores the particular relevance of ecological validity that fits in well with the ethnographic stance of the methodology. Finally it deals with the transferability of research findings across other similar environments.

2.6.1 Role of the Researcher

The role adopted was that of a formal role as a pure researcher, external to the project environment. The researcher was contracted out by the university to the Regional Government Department to undertake an investigation into the New IT System and fulfil a set of 5 deliverables - these have been achieved. As a consequence the researcher utilized the New IT System as a case study for a doctoral research project that was funded by the Department involved.

The researcher's role was initially tentative due to the unfamiliarity of the project environment and to the agricultural context. The primary ethnographic activity was participant observation that involved both formal and informal aspects of the researcher's role. For example, the formal role of observation through attending
meetings and workshops, and an informal role of observing spontaneous conversations within the project arena. These roles are differentiated in terms of their degree of participation and observation.

Adler’s Typology (Adler and Adler 1987 cited in De Laine, 2000) puts forward three membership roles within ethnographic research – peripheral, active and complete, where peripheral refers to the relevance of circumstance or issues of male/female gender, active reflects a researcher who is involved in and has responsibility in the central activities, and complete defines a commitment to the values and goals of the organization. If placed on a continuum this researcher’s role would be positioned between the peripheral and active. It represents a peripheral association within the agricultural nature and a participative involvement within the project arena that was not an active input into the central activities of the project development. This is supported by Walsham (1997) who puts forward two different researcher roles, firstly that of outside observer, and secondly that of involved researcher. The researcher was effectively a ‘fly on the wall’, an outside observer, no action research was involved.

Additional to the role of the researcher was the need to maintain appropriate boundaries between the researcher and subjects in order to avoid ethical problems arising from differing loyalties and expectations of the two groups of interviewees – the developers and the organizational people. This refers to maintaining neutrality and was exercised through the researcher’s refusal to disclose input by others when asked. This also served to uphold anonymity and confidentiality and promote trust.
Within the researcher's role there was also a duty not 'to spoil the field', in other words to protect interview subjects from bad experiences, such that potential consequences for future research projects were not created (Silverman, 2001).

2.6.2 Access to the Project Environment.

Entry into the project arena was sought and gained through the Project Board of the New IT System Project. The researcher was granted access to the project environment and to the project community. The project community consisted of a core team of 50+ of the developers and the organizational people. The project arena is spread across four floors of a centrally located government building where both the developers and organizational people were co-located together for the project duration. Development teams were organised into small integrated groups that were specialist and subject specific according to need (see Section 5.1.9, p160, for further detail).

2.6.3 Ecological Validity

An important aspect of valid research is the ability to translate findings across similar research areas. Consequently emphasis was placed upon ecological validity, which fits in well with the ethnographic stance of the methodology, because it specifically relates to the extent to which conclusions might be generalised to comparable social contexts. This reflects external validity that relates to the extent to which these research findings can be extrapolated beyond the immediate case study scenario to other similar environments (Johnson and Duberley 2000, Gill and Johnson 1991).

More specifically ecological validity encompasses identifying causal mechanisms by drawing out the process of cause and effect inside a very rich data environment.
within a particular context. This therefore supports the argument that if the same processes can be identified and implemented within a similar context again, the same problems can be expected unless the method is adapted. Thus a ‘lessons learnt’ approach can be applied from a retrospective view within a subsequent development environment.

Consequently it was the intention of the researcher to enhance the credibility of this piece of research by producing quality data through the consistent and systematic use of robust and reliable data gathering methods and techniques. This was complemented by applying triangulation during the data analysis stage to ensure validity of the findings.

2.6.4 Triangulation

Literature defines triangulation as the use of two or more methods of data collection in the study of some aspect of human behaviour. It is the combination of methodologies in the study of the same phenomenon that should have greater validity and reliability than a single approach (Denzin 1970); the comparison of different data collection methods to see whether they corroborate each other (Silverman 2001); and ‘the use of different methods and sources to check the integrity of inferences drawn from the data’ (Ritchie and Lewis 2004, p43).

Triangulation of qualitative data sources was the strategy adopted to reduce any bias in the data and to guard against claims that the study’s findings are simply an artefact of a single method, source or the researcher’s bias (Patton 1990, Silverman 2001). Triangulation was applied during the data analysis and evaluation processes enabling the researcher to corroborate the data gathered and deal with the possibility of bias in order to validate the findings. It was
facilitated by the longitudinal scope of this project as its duration allowed for combining observation, interviewing, and documentary sources that will reduce reliance on a single method to ensure that this piece of research is not just a subjective response of the researcher.

Activities involved checking the consistency of different data sources gathered through primary and secondary data collection methods. The researcher achieved this by comparing and cross checking the consistency of the said data that was assembled at different times and through the different data collection methods i.e. observation data, interview data and project documentation. More specifically, by examining the reliability of what participants said about the same thing over the duration of the IT protect and by comparing the perspectives of people from different view points.

Additionally, triangulation was also applied through the interpretation of the data using the examination of multiple perspectives or theories. The aim of this approach was to examine the data from the perspective of the various stakeholder positions to investigate different theories that would cast findings in a different light.

2.6.5. Researcher Credibility

It is also important to establish the credibility of the researcher. This research study is in fulfilment of a PhD degree and as such consists of a full time commitment over 3-4 years. Patton (1990) emphasizes that the credibility of the researcher also extends to their philosophical belief in the research approach adopted. Consequently the researcher confirms that the belief in the philosophical research strategy chosen of the phenomenological approach, and the underlying
epistemology, that guided the research was the most appropriate methodology to achieve the research aims and objectives.

Although this is not a first piece of research it is the researcher's first experience of ethnography and as such guidance and advice were sought from the supervisory team throughout the research project.

The researcher has no previous familiarity or understanding of the case study environment that involved an agricultural context. Thus a neutrality and impartiality was applied without pre-dispositions of biases that could have affected data analysis and interpretation (Patton 1990). Competence is demonstrated through the verification and validation procedures used to establish the quality of the research findings.

2.6.6 Ethical Issues

Ethnography requires the researcher to get close to the phenomena under investigation whilst maintaining a suitable distance in order to facilitate validated analysis (De Laine, 2000). It involves, and even necessitates, an element of trust in order to remove feelings of distrust and ensure freedom of communication by participants. Therefore to deal with any potential ethical problems, assurances were sort from the Project Board at the onset that confidentiality and anonymity of the participants would be maintained and that the information gathered would not be used adversely or out of context by them.

Potential ethical issues arising from the direct observation activities were allayed by an introduction of the researcher and the research objectives at the start of the research study. Consequently all project members were made aware of the
researcher's position, rationale and remit of the study being conducted. Thus the researcher was accepted into, and as a part of the project environment.

Informed consent was central to the interviews conducted, the researcher ensured that all interviewees adequately understood the nature of the research being undertaken and its anticipated consequences. In this way it was voluntary, they could decide whether to participate or not (Silverman 2001). Respondent validation was instrumental in maintaining an ethical stance as it afforded the interviewee the opportunity to clarify, delete or amend data that was either sensitive or that had inadvertently 'slipped out' during the interview sessions. As covert observation was not employed it did not constitute an ethical issue.

Lastly, as previously mentioned, the researcher acknowledges potential consequences for future research. There is an ethical duty not 'to spoil the field', in other words to protect interview subjects from bad experiences, such that they become unco-operative in later research projects (Silverman, 2001).

2.7 Research Schedule

The key research activities as described above were formalised into a research schedule that is set out below in table 2.1. The research study involved an initial period of orientation and immersion into the project environment, culture and substance. This led to the identification and shadowing of key people, and an analysis of existing project documentation, which was followed by a period of direct observations, initial interviews and subsequent semi-structured interviews and discussions. Data analysis and review of research activities were continual processes. Data validation, triangulation and thesis construction, writing and editing were final activities. The literature review, initially started before project
immersion, was on-going throughout the project duration. These activities are shown in table 2.1 below:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>March / August</td>
<td>September / February</td>
<td>March / August</td>
</tr>
<tr>
<td>Orientation &amp; Immersion</td>
<td>Observation Period, Initial Interviews, Conversations</td>
<td>Semi-structured Interviews, Data Review</td>
</tr>
<tr>
<td>Shadowing Key Informants, Document Analysis, and Review</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Semi-structured Interviews, Data Review</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Literature Research and Review</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2.1  
Research Schedule

The schedule shown above did not consist of fixed timeframes but provided a flexible framework that was designed to evolve and change in line with the project environment.

2.8  
**Strengths and Weaknesses of Chosen Methodology**

A major strength of an ethnographic research design approach is that ethnography is one of the most in-depth research methods possible. It is particularly suited to providing rich insights into the human, social and organizational contexts of information systems (Myers 1999).

A recognised disadvantage associated with ethnography is the length of time needed to complete the fieldwork, analyse the material and write it up. However this was not a major limitation because this piece of research involved a longitudinal investigation of a unique case study over three years. Additionally the unstructured data collection methods and focus on analytical induction may create problems in reliability and replication, but this can be counteracted by the application of ecological validity that enables findings to be transferred across similar environments (Gill and Johnson 1991), as was the case for this research study.
A further area of concern reflected the researcher's ability to challenge 'assumptions' made or held such that the work could be accused of having a lack of breadth. This is because the research concentrates on one organization therefore making it more difficult to develop general models. This criticism can be answered by Yin (1994), who argues that because it is possible to generalize from one case study to theory then it is equally possible to generalize from one ethnography to theory. However, in the context of this particular project any generalization is limited to the organizational contexts involved.

Vulnerability exists in the uniqueness of this project because unlike a questionnaire that can be repeated by another researcher to validate the conclusions, replication of this event is not possible. However, due to the strength of its in-depth analysis, its application creates a clear reliability that is built into the methodology. Thus the richness of the data gathered and its presentation is aimed at convincing the reader that had they been there they would have come to a similar conclusion.

Additional vigour through respondent validation also served to offset any observer bias imposed, whereby key participants confirmed that interpretations that had been understood by the researcher were a fair reflection of what was meant.

An acknowledged weakness was the inexperience of the researcher in the field of ethnographic studies. However prior to integration in the project environment the researcher investigated and examined this qualitative research methodology through a literature review and conversations with experienced ethnographers. Thus the researcher believes that this, combined with the guidance and advice
sought from the supportive and knowledgeable supervisory team, was sufficient in resolving this issue.

2.9 Summary

A qualitative research philosophy using a conventional ethnographic approach of interpretive research was adopted for the research study. Data collection concerned both primary and secondary data sources within a case study scenario that involved multiple and diverse data gathering techniques. The approach chosen was aimed at producing an understanding of both the context of the information system (IS) being developed and the process in which the IS influences and is influenced by its context. This refers to a broad view of the organizational environment of the IS and the wider external context within which the information system is related.

Care was taken that the research approach chosen was relevant and suitable to the IS field, would satisfy the research aims and objectives, and was appropriate to the nature of the problem to be explored. It further considered the constraints of the available time and resources, and was believed to be sufficiently rigorous to enable validated conclusions to be drawn.

The suitability of this approach was further reflected by its association with IS research where there are many instances of this research style being adopted within IS field by authors such as Beynon-Davies et al. (2000), Myers (1999) and Walsham (1997).

Data was gathered and analysed across the project population where the same people were interviewed recurrently in an iterative process that is associated with
grounded theory (Glazier and Strauss 1967, Strauss and Corbin 1990). Consequently data were not taken singularly but involved the comparison of information not only relating to the same phenomenon, but which derived from different phases of fieldwork and accounts of different participants to facilitate triangulation. Thus the researcher believes that ecological validity has ensured the quality of subsequent research findings.

The researcher believes that the data collected and analysed are valid and that the research design has addressed the potential biases, problems, constraints and limitations.
3. Introduction

Late entry into the project environment caused by funding problems, and the need to obtain security clearance for access into the project arena enabled the researcher to focus on initial literary review activities early on. These were supplemented and extended throughout the project duration. As set out in the research methodology, primary and secondary material were gathered through a diversity of means. It involved both traditional and electronic media across academic and practitioner sources. This chapter sets out the literature review for the research study. It aims to present an understanding of the current status of RAD, and its position within the field of IS.

There are basically three types of evolution that exist within the IS arena that need to be explored in order to provide a foundation for an understanding of the current status of RAD, such that the reader is able to fully appreciate RAD’s emergence.

Firstly, it is necessary to present a brief history of the development of Information Systems and the shift in organizational practices to illustrate how this has affected the system development arenas. Secondly as the history of software has close links to development environments, it is therefore important to provide an overview of the evolution of development environments and associated technological advances. Thirdly an overview of advances in development methods is essential to present some discussion on the progression of system development
methodologies such that a holistic understanding of these phenomena can be achieved in relation to RAD.

From this analysis a genesis of RAD can be presented together with a current context of its application as a development approach. This will enable the identification of development problems associated with RAD with particular reference to project size, and where identifiable, with specific reference across large and complex development arenas. A corollary will be the creation of a table from the opinions and views expressed in the literature that can be compared to, and extended by a similar table derived from the analysis of the research case study of a large and complex RAD-type development project. This comparison will allow the researcher to present a model of critical success factors associated with a RAD-type development that extends its utility beyond the small to medium sized projects recorded in the literature. Thus it will address the recognised knowledge gap within current literature, contribute to the current debate, whilst also responding to the key research questions of RAD’s scalability for such development projects.

3.1 Brief History of the Development of Information Systems

In a recent study Bannister (2003) argues that by considering the history of Information Systems (IS) from Charles Babbage’s Difference Engine (1821) to the launch of IBM 360 family of computers (1964) an insight can be gained into how the temporal context of IS impacts on their development.

The early stages of computer use were dominated by military and government purposes. Consequently system and software development were primarily controlled by the scientific paradigm of specialist mathematicians and scientists. It
was not until numeric machine code programming was replaced in the 1950s with high-level languages using a more symbolic representation (such as FORTRAN, COBOL, Pascal) that systems development moved into the business environment. Bannister (2003) proposes that the first real commercial computer system application of electronic computing was the LEO computer system (Lyons Electronic Office), a payroll system implemented by John Pinkerton in 1951 (this view is confirmed by Bird 1994, and also by Caminer et al. 1997, Caminer 2001 cited in Bannister 2003), which led the way for computerised business information systems.

Thus it was during the 1960s that the computer began its role in the business community assisted by the development of new programming languages such as COBOL (Common Business Oriented Language) that speeded up the development process, enabling the first generation of on-line systems for transaction processing to emerge. This development meant that larger types of problems could be solved leading to larger scale and more complex system developments that were constrained only by the hardware available. Gradually, as the hardware restrictions to systems development were being resolved the software constraints became visible and development methodologies evolved in an attempt to control the development process.

The escalating power and accessibility of hardware resulted in an increased sophistication of development methods, techniques and tools that became available for software development. Hence Information System application development moved away from highly specialised individual developers to the concept of formalised development teams using appropriate development methodologies.
### 3.1.1 Information Systems Development Themes

Avison and Fitzgerald (1995) have identified four themes throughout the transition of IS developments. Firstly, holistic systems development from the organizational point of view. This concerns developing IS for an organization as a whole rather than for its individual functions regardless of the methodology adopted, where computerization was initially aimed at increasing efficiency and reducing labour costs rather than the changing nature of the business. Checkland and Scholes (1990) later developed this concept into the Soft Systems Methodology approach that attends to the fuzzy aspects of undefined system activities in contrast to the hard approaches of structured design and data analysis representing a single viewpoint.

The second theme introduced a process modelling approach that was associated with disciplined development methods and places the emphasis on systems analysis. It focused on data modelling that decoupled the system logic from the physical data so that the data is independent. Data was seen to represent the fundamental building blocks of a system.

It was the changing nature of business environments together with the adapting profile of systems development environments that raised a concomitant need for more rapid development of systems with frequent tangible results to meet short term needs (Fitzgerald 1996). The fundamental structure of an IS changed with the introduction of the database approach. Powerful back-end Client/Server databases such as Oracle and SQL Server emerged replacing centralised mainframe databases which had themselves replaced filed backed storage systems. As popularity of the Client/Server paradigm increased, and the volumes
of concurrent users grew, Client/Server architecture evolved to optimise the execution of data management and support the design and implementation of distributed systems that enabled system expansion in terms of extensibility and scalability (Boss 2002). It provided developers with the flexibility to seamlessly integrate new technologies that responded to the evolving modular, dynamic and user orientated IS environments.

The third theme reflects the use of automated software tools in IS development that deal with repetitive or rule based elements. Prototyping evolved with the availability of software tools, in particular 4th Generation Languages (4GLs) and application generators that reduced the cost of system development and shortened the project lifecycle. At this time engineering principles were introduced to fulfil both the growing demand for larger and more complex programs, and the increased potential of hardware that had not been exploited fully by advances in software development. Mathematical precision was used in specification and design so that the solution could be proved to work. Jackson Structured Programming (JSP) is an example of this.

Lastly the fourth theme revolves around the increase of user participation in the development process. Users and analysts work as part of a team throughout a project, for example in Joint Application Development (JAD) workshops to elicit more meaningful feedback, in conjunction with prototyping so that the correct system is built. Thus system development methodologies such as the RAD development approach and the DSDM framework evolved which encapsulated the benefits of user involvement, JAD workshops and prototyping whilst operating within an iterative and incremental cyclic development structure.
3.2 Evolution of Development Environments

Businesses communicate in an inherently asynchronous and distributed manner. Information systems developed to support these environments have evolved inline to reflect this asynchronicity. In fact advances in system development environments seem to imitate the characteristics of distributed human collaboration through system IT architecture, services and protocols that is representative of human behaviour, for example like that of Client/Server systems.

3.2.1 Monoliths to PCs

A major influence on the evolution of system development methods was the escalation in technical advances of computer power and system architectures. In the 60s and 70s large centralised monolithic host computer systems were needed to support the hierarchical business environments of that era that focused their functionality and software on one centralised powerful machine. Bureaucratic organizations deployed technology to address specific business needs. Consequently IS systems concentrated primarily on mainframe host computing using ‘dumb’ terminals dependent on the host for their functionality. Users interacted with the host mainframe through the ‘dumb’ terminals that relayed information back to the host computer. However, although the mainframe environment of the 60s / 70s provided powerful computing power, system development continued to be maintained within the technological arena (Graham 1989).

These practices led to a situation where systems development was characterised by isolated pockets of technology, applications and information that were host specific supporting a diversity of system software associated with inconsistent and
low performance. During the 70s and 80s due to recognised problems with system development environments a focus emerged on socio-technical systems that aimed to optimise both technical and social aspects of systems design. The resultant plethora of vendor proprietary hardware and software products designed for these specific computer systems led to a customer lock-in of incompatible technology and increased isolation of host networks that was very constraining for system developers (Tapscott and Caston 1993). In response to this constraint, new types of computing architectures arrived to challenge the host specific style of computing operations, and address the growing complexity of the scope and nature of business operations.

The advance to PCs and Desktop Computers with increased capacity replaced the 'dumb' terminals and fuelled the move towards sharing/splitting the processing demands between mainframe and the PC. This advance drove the shift towards downloadable file architecture from the mainframe to individual personal computers (PC) enabling users to manipulate applications through Graphical User Interfaces (GUIs).

The move from mainframe to applications with responsive interfaces on PCs was seen as a way to maximise the effectiveness of the users. Shafe (1995) aligns this with the need for a new approach to application development process and the move towards prototyping, JAD and the introduction of 4GLs that served to reduce development costs. GUIs provided consistency across applications via easy to use interfaces that is associated with a change in the role of the user and the redefinition of business processes. Furthermore it is important to also consider the developments of operating systems that changed in line with technological
advances. The move away from DOS to evolving Microsoft Windows environments in the early 80s i.e. 3.1 OS/2, 95, 98, XP.

Associated problems with islands of users, applications and information combined with the difficulties of integrating across proprietary systems fuelled a new approach to system development architecture. Attention was applied to both integrating the previously disparate applications of technology through distributed networked infrastructure based on standard components and interfaces; and on creating architectures capable of supporting the inter-working of various components (Tapscott and Caston 1993).

3.2.2 High Level Development Environments

The lack of portability between different computers led to the progress to the high-level development environments of the 3rd Generation Languages (3GLs) and 4th Generation Languages. Programmers using 3GLS and 4GLs were able to ignore many of the low-level details of the previous machine languages of the 1st and 2nd Generation Languages. These required, respectively, the writing of long strings of binary numbers (0s and 1s) to represent such operations as 'add' and 'subtract' and, simple mnemonics such as A for 'add' or M for 'multiply' which were translated into the machine language by a computer program (an assembler). Both of these languages were time consuming and labour intensive and often error prone (Vonk 1990). It was recognized that the closer the syntax, rules, and mnemonics of the programming language could be to 'natural language' the less likely the programmer would be to make mistakes and inadvertently introduce 'bugs' into a program (Vonk 1990). 3GLs such COBOL, Pascal, BASIC, and C are procedural languages that required the programmer to state what actions must be executed, and are translated into machine code by a program called a compiler or
interpreter that performs a large amount of the work. Whilst the 4GLs concentrate on the nature of the problem rather than on the procedural way it is solved, they specify what is to be accomplished without describing how. Furthermore they were also recognised to have facilitated rapid development and therefore prototyping (Vonk 1990), 5th Generation Languages (5GLs), an outgrowth of artificial intelligence (AI) research, are still in their infancy.

3.2.3 CASE (Computer Aided Software Engineering) Tools

One area that has progressed to provide assistance with high level development environments is the evolution and availability of software development tools. CASE tools provide support across stages of the System Development Lifecycle (SDLC). Their purpose is, through the automation of part(s) of the development process, to raise levels of productivity and quality (King and Galliers 1994).

RAD development requires the support of rapid development tools such as CASE tools that are commonly used to cover many of the development tasks from business modelling to requirements capture and implementation. A distinction is sometimes drawn between Front-end² CASE tools which focus on the investigation, analysis and design stages, and Back-end³ CASE tools that are associated with code generation and implementation activities (Vonk 1990). Front-end CASE tools do not perform any analysis or design but rather they enable modelling techniques to support systems analysis and design to be implemented and integrated to allow cross referencing. For example conceptual data modelling i.e. Entity Relationship (ER) models, Data Flow Diagrams (DFDs).

² Front-end CASE tools also known as Upper CASE tools.
³ Back-end CASE tools also known as Lower CASE tools.
Benefits of using CASE tools included interactive systems development that increased acceptance of the system, increased productivity and perceived quality whilst reducing system maintenance. Disadvantages associated with CASE tools are represented by the cost and risks involved (Vonk 1990). Most CASE tools are methodology driven and therefore require use of a structured methodology. Avison and Shah maintain that 'the implementation of both CASE and new methodology at same time compounds risk of failure' (1997 p272). A further disadvantage was that the developers tend to change requirements to suit the technology of their application during construction (Ljubic and Stefancic 1994).

The concept of CASE is broad and it includes compilers, project management tools, and even editors (Tolvanen 1998). CASE tools are a set of programs that assist analysts, software engineers and programmers during all phases of SDLC, and enable emphasis to be placed on analysis and design. They were perceived as a means to increase the speed of development such that a system could be produced before business needs changed, and also importantly as a means to increase competitive advantage.

Previous research by King posits that the 'use of CASE tools can help to deliver software quickly, with few defects...' (1996, p181). This view corresponds to Martin's original thinking that there does not have to be a compromise between high speed and high quality, they 'go hand in hand' (1991, p5) providing the development methodology is appropriate.

Tools most commonly associated with RAD include visual development environments such as Microsoft's Visual Basic. Similar environments exist for other languages such as Java and C/C++. The common feature is that the
developer 'draws' the interface and then supplies code to add functionality. This makes them ideal prototyping tools, as linked interface screens can be created quickly and easily. Once these screens are accepted, the required functionality can be added.

3.3 Overview of System Development Methodologies

Early systems development was hampered by a lack of structure and discipline, relying on the skills and experience of individuals, whose approach was often ad hoc and haphazard. There is some consensus that the evolution of system development methodologies can be aligned directly to hardware and software technological advancements and their increased sophistication (Osborn 1995, Elliott 1997).

It is further suggested by Osborn that the evolution to Client/Server applications influenced the use of layered hardware architectures and distributed database infrastructure, and that system development approaches such as RAD were used to develop application focused front ends to architectural solutions. The progressive development of hardware and the emergence of software constraints emphasized the need for a more methodical approach to system development. Existing literature acknowledged the need for new IS development practices brought on by the shift in developmental constraints from hardware to software processes (Martin 1991, Weaver et al.1998).

Consequently software engineering became an important mechanism in the systems development process that aligned IS development with the use of the more structured framework of a Systems Development Lifecycle (SDLC) i.e. a schedule of phased sequential stages. See figure 3.1 below:
As the above figure illustrates this traditional SDLC follows a rigid sequence of steps, each of which requires formal sign-off on completion. A complete, detailed requirements analysis is conducted in the early stages of SDLC that attempts to capture the system requirements in a Requirements Specification such that user involvement was confined to these early stages. This stage was followed by a complete system design, and subsequent development and testing (McConnell 1996). Problems experienced reflected the fixed status of the requirements gathered early in the development process that the developing system did not accommodate any changes in business needs. However, initial efforts to overcome the problems of traditional development models influenced by hardware and software considerations were directed towards the implementation end of the lifecycle. The Waterfall Model was the most widely used.
3.3.1 Waterfall Model

The oldest and best known SDLC is the Waterfall Model that emerged as the dominant model during the late 60's, and was first reported on by Royce in 1970. It axiomatically falls within the traditional lifecycle paradigm. Similarly it concentrates on specifying requirements up-front before development begins, followed by several logical steps to schedule the project see figure 3.2 below. It was documentation driven and became the standard software development model, and is commonly referred to as the 'traditional' development model. The main characteristic is that of a linear lifecycle that resembles a generic model for software development.

![Waterfall Model Diagram]

Figure 3.2 Waterfall Model

It is argued that although it does not provide product visibility or tangible results until the end of the project lifecycle, progress can be monitored through its rigorous documentation. It was suited to development projects with stable requirements, where the technical methodologies were well understood, where errors that are identified in the early requirements gathering stage occur during the lowest cost phases (Martin 1991, McConnell 1996).
However, this monolithic model that regarded development as one large process came under attack and increasing criticism due to its rigid design, inflexible procedure, inability to accommodate uncertainty and limited user involvement (Graham 1989, Martin 1991). Additionally, it does not yield a working version of the system until late in the process when late changes are both costly and complex. Real projects rarely follow the sequential process flow that the model proposes, and the high levels of system development failure recorded during the software crisis of the 60s/70s were attributed to these problems. SSADM was developed to present a more formalised approach by imposing a more rigorous discipline within the design and development processes of the SDLC.

3.3.2 Structured Systems Analysis and Design Method (SSADM)

SSADM is a development method whereas the Waterfall Model is a model of systems development stages. In its early stages the SSADM incorporated many of the tenets of the Waterfall Model. The Structured Systems Analysis and Design Method (SSADM) launched in 1981 was an attempt to reduce systems development failure by imposing a more rigorous discipline within the analysis and design process phases of the structured lifecycle. It does not cover the entire SDLC.

SSADM distinguishes between the logical and physical views of an IS, taking account of not only the functional and data aspects of the IS, but also their associated dynamics to produce a greater understanding of the system requirements (Weaver et al. 1998). In 1983 it became a mandatory approach for all UK Government IS projects and was recognised as the de facto industry standard, however it is no longer mandated by government. SSADM incorporates a vast amount of documentation and is regarded as a heavy methodology,
compared to light methodologies such as RAD and DSDM, that are associated with reduced documentation and management effort. Figure 3.3 illustrates how SSADM aligns to the early stages of the SDLC.

Figure 3.3 SSADM and the System Development Lifecycle
Source: Adapted from Weaver et al. (1998)

However whilst the move from static to dynamic systems design improved the discipline of systems development it did not resolve all the problems it was expected to solve (Graham 1989). The associated problems with up-front requirements elicitation and the late identification of major problems continued to result in a high percentage of project overrun, over budget failures and systems that did not meet user requirements (McConnell 1996, Coughlan and Macredie 2002, Boehm 1999). Consequently growth and change were recognised as intrinsic elements of IS development thus creating a demand for other more flexible development approaches such as Boehm’s Spiral Model (Lycett and Paul 1997, Fitzgerald 1996 cited in Avison and Shah 1997) this is further discussed in Section 3.4.1, p79.
The increasing complexity of IS developments, facilitated through technological advances and the evolution of the development environments fuelled the move towards modular development, such that large, complex projects could be broken down into separate development modules through incremental and evolutionary approaches to development.

3.4 Modular Development

The recognised need for increased flexibility and the evolution of systems development in terms of size and complexity promoted the move towards modular development. Modular development incorporated incremental development and incremental delivery processes as alternatives to the monolithic lifecycle. There is a distinction between incremental development and incremental delivery. Incremental development breaks down a project into development modules or units that can be delivered as a complete system. Incremental delivery, which is also known as evolutionary development, refers to the development and release of developed module(s) or unit(s) where feedback from the 'first' module or unit is looped back into the subsequent module or unit being developed. It is possible to adopt incremental development without incremental delivery. For example, incremental development, such as the build and test model, implements the final stages of system development, after finalization of the design stages, through a series of smaller blocks (Graham 1989). Whereas an evolutionary approach applies a complete development process to each of the sequential stages of the SDLC, like a series of mini waterfalls. In each case both are iterative processes where prototyping is integral to reducing the risks associated with the traditional linear concept.
The breakdown of complex projects into smaller, more manageable components was seen as the way forward for large, ambitious projects that carried a high risk of failing to meet some, or all, of the desired goals. In this way projects become more manageable, easier to specify and implement, and more able to respond to changes in technology, thus offering greater control (Government Review 2000). Modular development identified two significant dimensions to large IT-enabled business change projects, firstly the range of business functions that they support, and secondly the level of support that was offered to business processes. The idea was to apply both a modular, and an incremental approach, see figure 3.4.

The modular approach facilitates the development of ‘parts’ of a large project that underpins a limited set of business processes, capable of providing their own value without the need of other parts of the project. The incremental approach begins with a component of the overall system that is of limited functionality, but then builds upon it to increase its value to the organization. It is felt that this provides a higher probability of success if, rather than aiming to supply the complete range of business support functions, smaller projects are designed to deliver those ‘parts’ that can be separated out into single modules.

Figures 3.4 and 3.5 below illustrate the different concepts of modular delivery and incremental delivery. As can be seen modular delivery focuses on the ‘A’ range of business functions. The incremental model divides the product into builds, where sections of the project are created and tested separately (i.e. A,B,C, & D). When combined, a modular and incremental approach reduces potential risk.
The continued growth and change of system development projects within the IS arena emphasised a need for other more flexible development approaches such as Boehm's Spiral Model (Lycett and Paul 1997, Fitzgerald 1996). This is discussed next.

### 3.4.1 Boehm's Spiral Model

Boehm's Spiral Model, developed in 1988 attempts to map the development spiral timeline to a linear timeline. It incorporated the best features of the Waterfall Model coupled with prototyping as a process model, and introduces risk assessment as part of the development process. It develops smaller parts of the IS allowing results to be prototyped earlier in the SDLC. This facilitates feedback review early in the project lifecycle after each iteration/spiral, see figure 3.6 below. The Spiral Model is risk oriented, where risks are addressed through spiral iterations until resolved. An initial version of the system is developed and prototyped, then continually modified based on the feedback received.
Boehm's spiral is a complex figure where each iteration involves six steps (in bold on the out edges of the spiral) starting in the centre and working outwards that provides increasingly complete versions of the system being built. Thus with each spiral the risk is evaluated, assessed and plans put in place to deal with it. The disadvantage is that it is complicated and necessitates rigorous management. Although the Spiral Model incorporates prototyping and risk analysis, it cannot deal with unforeseen business changes (Easterbrook 2004).

There have been several variations over the years but Boehm's Spiral Model remains the best known. Other examples include the Fountain Model, Tornado Model, Whirlpool Model and the V – W Stage Model. A corollary of these methods gaining popularity was the rise of prototyping as an alternative to the traditional lifecycle for information system development (Alavi 1984), which led the move towards the visual development environment of Rapid Prototyping Models.
3.4.2 Rapid Prototyping

A prototype is a medium around which users and designers can negotiate and share an understanding of the system (Tudhope et al. 2001). Vonk states that a prototype is "a working model of (part of) an information system, which emphasizes specific aspects of that system" (1990, p20). Prototypes can reflect a vertical or horizontal nature, where a vertical prototype contains both high-level and low-level functionality for a restricted part of the system i.e. a narrow slice that covers only one specific aspect of the system to provide complete depth for that aspect. Whilst a horizontal prototype shows a broader view of a system but has limited functionality, i.e. covers a large breadth of features and functions rather than depth (Preece et al. 1998).

A general interpretation considers prototyping to be physical mechanism designed to highlight and identify potential problems with system functionality and any mismatch to user expectations. Therefore prototyping is linked with user satisfaction, it is a 'what you see, is what you get' approach affording users the opportunity to provide meaningful feedback early on in the systems development in order to prevent costly redevelopment later in the project lifecycle, as is the situation with the traditional Waterfall Model approach.

It is characterised by users and developers interacting together in a collaborative manner throughout the design and development processes. Figure 3.7 below represents the nature of the prototyping process that occurs during the design/development processes.
In her comparison between prototyping approaches and the more traditional development lifecycle, Alavi (1984) advises that prototyping facilitates communications between users and designers promoting more meaningful feedback. This view is supported Dearnley and Mayhew who believe that the ‘use of a prototype fosters the generation of a ‘team spirit’ between users and analysts’ (1983, p38). Additionally they inform that the involvement of users in prototyping activities affords better access to their knowledge and understanding, providing more meaningful input and feedback. The need for frequent interaction between the users and developers serves to break down potential communications barriers thereby preventing possible conflict situations that can arise when users are not involved. However Alavi points out that rapid prototyping could also be problematic in terms of controlling scope, particularly with large IS.

Since that time prototyping has become an important element in system development projects and is commonly applied. Maner (1997) sets out a broad description of the different prototypes outlining the benefits and drawbacks.
Stapleton (2002) later confirms the importance of prototyping as it enables continuous negotiation of system requirements.

Prototyping was seen to address the communication gap between developers and users during requirements elicitation previously experienced in the more traditional development approaches (Vonk 1990). It requires a high degree of user involvement and therefore is more likely to meet the users needs, thus engendering a higher acceptance of the eventual system. It provided users with a tangible means of understanding and evaluating the system being developed, as well as presenting developers with more meaningful feedback, and introducing a baseline for progressing development (Alavi 1984). For developers prototyping provided a milieu to test their understanding of the IS environment and its changing business needs, whilst providing an analysis test bed and a vehicle to validate and evolve system requirements. Importantly it also provided an opportunity to ascertain what is actually feasible within a project’s technological constraints (Cross 1998).

Consequently it gained popularity as an alternative to the traditional SDLC for information system development (Alavi 1984), and this acknowledgement led the move towards Rapid Prototyping Models that evolved from prototyping approaches (Humphrey 1989, Kan 1995). Although many prototyping models have been occasioned, there are two generally accepted forms. Firstly throwaway prototyping that is evaluated with users and subsequently developed in a different environment, or which employs a different programming language. Secondly evolutionary prototyping, also referred to as incremental development, where users are able to experiment with each increment, the prototype is evaluated and refined, eventually evolving into a part of the final system (Crinnion 1992,

A general view put forward posits that evolutionary prototyping/rapid prototyping is generally associated with iterative development (Martin 1991, DSDM 2001, Whitely 2004). It is particularly suited to situations where the requirements are dynamic and not well understood (McConnell 1996). It is the evolutionary prototyping approach that is associated with RAD and thus is further explored.

3.4.3 Evolutionary Prototyping

Evolutionary prototyping is an approach to system development where an initial prototype is produced and refined through a number of versions to the final system. The objective of evolutionary prototyping is to deliver a working system to end-users and is particularly suited to user interface systems. It is associated with phased deliverables, i.e. building the system in an iterative way, utilising a cycle of inspection-discussion-amendment until the user is satisfied with system (Beynon-Davies 1998). It is based on techniques that allow rapid system iterations that demonstrate the adequacy of the system.

Evolutionary prototyping reflects a lifecycle model that develops the system throughout the project (McConnell 1996). Parts of the system are demonstrated to the clients and refined in light of feedback received until both developers and clients agree that the prototype is ‘good enough’ and it is released as a final part of the system. It is of particular benefit in circumstances where the business requires change rapidly. However the literature informs that planning and control of prototyping projects are more difficult as it is not always possible to estimate
development time because the form of the evolving system, some of the users needs and the number of revisions to the prototype are unknown at the outset (Alavi 1984, McConnell 1996). Additionally, there are drawbacks where existing management processes assume a Waterfall Model of development that is unable to respond to continual changes. This scenario tends to corrupt the system's development structure necessitating long-term maintenance that increases cost.

In spite of the perceived advantages of prototyping, literature reports that initially rapid prototyping had very poor performance characteristics, which made it unsuitable for large or complex systems (Goa 1998). It was also regarded as undesirable where systems featured large numbers of concurrent users, or large volumes of stored data. Consequently it was mostly applied to small systems development.

An inherent weakness was a tendency to avoid documenting system requirements formally. A criticism voiced by Oz (2002) that prototyping increases a risk of incompatibility across parts of the system, is shared by Osborn (1995) who states that because prototyping primary focuses on visual modelling, it supersedes both written specifications and data models, thereby introducing an increased risk of skewing systems specifications. A corollary of this lack of documentation is that it may lead to the wrong system/problems being focused upon (Necco et al. 1989). As system developments continued to fail, non-perfect solutions were often accepted if they are delivered on time. Most customers will accept 90% of a system in exchange for getting the product faster. Hence RAD evolved as a means to address some of these issues (DSDM 2001, Raffoni 2000).
3.5 **What is RAD?**

The above sections have presented the background and context for RAD. The aim of this section is to clarify for the reader the essence of RAD. It looks at Martin's fundamental philosophy, identifies and describes the main characteristics, and RAD's relationship to the DSDM approach that is perceived to have evolved to promote a non-proprietary public domain version of a RAD framework. It further maps the original features as defined by Martin against the DSDM 9 principles that are used to evaluate the presence of a RAD approach within the case study scenario.

3.5.1 **RAD Defined - The Essence of RAD**

In 1991 Martin describes RAD as a method *'that uses computerised tools and human techniques in a tightly woven fashion to achieve the goals of high speed, and high quality'* (1991 p6). Since that time RAD has been described in relation to its application as an approach or framework, as a method, or as a set of techniques and tools. Additionally there has been some overlapping use of the terms 'methodology' and 'method', therefore it is important to define these terms in order to maintain a consistency throughout this research study.

Martin's original definition (1991) is that of RAD as a methodology - *'the set of tasks that have to be accomplished in order to develop a system'* (1991, p6). An understanding of 'method' that is put forward by Sadler *et al.* (1996) and Beynon-Davies (2002) is that of a process model that defines the tasks/steps to be performed, the order in which they are performed and how they should be undertaken in a given development process. For consistency within this piece of research, the researcher acknowledges the differences in terminology and takes
the terms 'method' and 'methodology' to be the same i.e. the researcher treats them as synonyms.

A technique is 'a means of carrying out one operation' Martin (1991, p6), who puts forward data modelling as an example. This concurs with the meaning by Beynon-Davies in 2002 who says that a technique is 'some systematic activity within the development process' (p593). Beynon-Davies further describes a tool as 'some software used to aid the development process' (2002, p594).

Additionally the term 'framework' is a more contemporary expression. There is no singular definition reported across the RAD literature but a general interpretation. The use of the terms 'framework' and 'approach' are classed as the same. For example, the DSDM consortium conclude that RAD is more of a framework than a method and describe it as high-level project management approach.

RAD is an iterative and incremental development approach that compresses the analysis, design, build and test phases of the SDLC into short, iterative development cycles. Thus is differs from the traditional development approaches such as the Waterfall Model. Figure 3.8 below illustrates how the iterative nature of RAD deviates from the linear, sequential style of the Waterfall Model. It is these development cycles that are able to accommodate the growing uncertainty and increasingly volatile nature of current development environments. This type of approach necessitates the collaboration of small, diverse teams of developers, end-users and other stakeholders working to tight deadlines in an effort to optimise speed, unity of vision and achievement of the development process. (Martin 1991, Beynon-Davies et al. 1996 1999, 2000, Elliott 1997, Tudhope et al. 2001).
Martin first described RAD as a development lifecycle designed for high quality systems with faster development and lower costs, designed to address the problems of lack of speed and quality associated with the more traditional development approaches. RAD, as a term, is used to describe development projects that emphasize speed. The word 'rapid' does not denote a quick fix but rather refers to an incremental and iterative delivery process. Since that time RAD has been applied as a development approach/framework, a development method and/or development components dependent upon its purpose; this threefold approach to RAD is further discussed through the debate presented in Section 3.7.2.1, p104. It is a development approach that necessitates a planned infrastructure in terms of information and technical architectures (Martin 1991) as shown in figure 3.9 below:

Figure 3.8  The Waterfall Model and RAD Development Approaches
Source: www.credata.com

Figure 3.9  RAD within a Planned Infrastructure
Source: Adapted from Martin (1991)
He put forward four essential, interrelated factors which underpinned the RAD approach i.e. People, Management, Tools and Methodology (1991, p10), this is illustrated in figure 3.10 set out below:

![Diagram](image)

Figure 3.10 The Four Essential Aspects of RAD
Source: Adapted from Martin (1991)

This would suggest that from the embryonic stages of RAD, there was recognition of the need to involve appropriately skilled people, and to employ tools and techniques within an effective methodology, that involved strategic project management i.e. of both the people and the processes throughout the project lifecycle. Views expressed in the IS literature confirm that Martin’s original interpretation, to some extent, still applies today, but that increased focus is placed upon the user interactions and activities that currently play a more central role within a project development, rather than emphasising people in terms of the technical skills needed and training (Beynon-Davies 1998², 2000, Bates 1995, DSDM 2001).

It is these four factors that Martin suggests when integrated achieve high standards of quality. This combination of high speed, high quality and low cost systems development promoted quality development (see figure 3.11 below).
Unlike traditional methodologies, RAD offered a platform independent development framework that was able to accommodate various methods, tools and techniques such that it provided a synergy for the planning, data/process modelling, and prototyping of a RAD approach. The rapid nature of development meant that it was able to respond to the increasingly dynamic nature of evolving business environments (Martin 1991, Elliott 1997, Raffoni 2000).

3.5.2 Types of RAD Project

RAD projects are sometimes distinguished in terms of intensive and non-intensive forms. A non-intensive approach refers to projects where system development is spread over a number of months involving incremental delivery, compared to the intensive RAD where project personnel are closeted away to achieve set objectives with a 3 - 6 week timeframe (Beynon-Davies 1999). The case study concerned in this research project is a non-intensive approach, that is considered to be more adaptable for larger projects because development can be organized into separate development modules/blocks for incremental development and phased delivery.
3.5.3 RAD Stages

Martin describes four phases of a RAD development approach, these are set out in figure 3.12 below. Briefly explained requirements planning provides a broad view of the system requirements based at a high-level. The user design stage concerns a detailed analysis of the business activities where users and developers collaborate within JAD workshops. It is this stage that utilises the I-CASE tools (integrated CASE tools). Prototyping is generally used to align the developers interpretation of requirements with the users actual needs. This stage is followed by the construction stage and although with a traditional development lifecycle it is a separate phase from programming, with an I-CASE toolset the relationship between design and coding is changed.

![Figure 3.12 Four Phases of RAD](source)

Figure 3.12 Four Phases of RAD
Source: Adapted from Martin (1991)

CASE tools enable faster design and development that requires close involvement of system users who validate the system as it is being built. In this way construction can proceed quickly to ensure that the delivered product meets the needs of users at cutover. Cutover involves comprehensive testing, training of users and implementation of the system test.
3.5.4 Characteristics of RAD

Martin (1991) characterised the main features of RAD as user involvement, prototyping, CASE tools, SWAT teams (skilled with advanced tools), JAD workshops and timeboxed development. Since then RAD's characteristics have been largely described dependent upon its purpose i.e. a development method, project management approach or use of RAD techniques and tools (Beynon-Davies 1999, Boehm 1999, Elliott 1997, Bates 1995). However, it is relevant within this section to explore further the elements of JAD workshops that incorporate the timeboxing, descoping and MoSCoW features (see section 3.5.4.2 p95), together with prototyping, as these are considered fundamental to a RAD approach.

3.5.4.1 JAD workshops

JAD is a systems development technique originally conceptualised by IBM (developed by Chuck Morris and Tony Crawford) in the early 1970s, and later further developed by James Martin (Elliott 1997). As such, like RAD, it evolved within the commercial arena with little influence from the academic world. Consequently JAD has had minimal theoretical basis, it has been driven by studies of group work and meetings, thus it evolved from behavioural underpinnings designed to achieve technical goals. Martin informs JAD 'produces results faster than the traditional writing of a specification' (1991, p156).

JAD workshops are utilised throughout the development process and are considered to be a key aspect of RAD, they form a fundamental part of the iterative development process. The essence is to get users involved in structured meetings to make timely decisions, around which development activities evolve. The integrated team environment is aimed at breaking down communications
barriers and increases levels of trust and confidence. This level of user participation is aimed at increasing quality, and reducing both costs and lifecycle time (Carmel et al. 1993). Potential benefits are described in terms of user commitment that engenders a synergy and consensus within the decision-making procedures.

Typically JAD sessions consist of small and diverse teams of 4-8 key people, comprising developers, users and other stakeholders empowered to make decisions for a rapidity of development within the short iterate cycles that optimise speed, unity of vision and purpose of the development process. Developers use CASE tools and modelling techniques to communicate and confirm to users their understanding of the requirements given (Martin 1991, King and Galliers 1994, Elliott 1997, Maner 1997, Saleh 2001). The recommendation is that they take place in ‘clean rooms’ away from business/developer environments, and are equipped with the appropriate support facilities i.e. flip charts, post-its, pens. In other words the workshops are free from interruptions from the work environment such as telephone calls. The aim is to produce documented business requirements within a set timebox (Beynon-Davies 1998, 1999).

However such sessions need careful management to ensure that the JAD groups are productive and reach a consensus otherwise their benefit is lost. Figure 3.13 shows how the JAD process fits into the iterative development of a RAD development approach. This has been aligned against the processes of the traditional linear approach for comparison.
JAD as a group based development approach provides a framework for more concentrated and enhanced user involvement within the systems development process than the SDLC approach. With RAD, project control is achieved by applying fixed timescales to project activities. For example, in JAD sessions when slippage occurs, requirements are prioritised and descoped to meet a specified timeframe sacrificing functionality if necessary. If slippage occurs then emphasis is placed on reducing and prioritising business requirements to fit the desired timebox, rather than by extending the deadlines. Thus the development team is able to focus on the pieces of functionality that have the highest business value such that the system is considered 'fit for business purpose' at delivery. Consequently secondary requirements are dropped or left for later development releases in line with the MoSCoW concept (see Section 3.5.4.2, p85). The timebox deadlines are not extendable, they are immovable, thus it is the functionality of system that is descoped in order to achieve development schedules (Martin 1991,
Maner 1997, Beynon-Davies 1998\(^1\)). The focus is on delivery within a timebox, scope can change but delivery dates cannot be extended.

### 3.5.4.2 MoSCoW, Timeboxing and Descoping

Business objectives are prioritised and grouped into one of four categories that are often referred to as the mnemonic MoSCoW, for example:

- **Must Haves** - critical system factors
- **Should Haves** - cost effective direct benefits
- **Could Haves** - the nice to haves, no direct benefit to client
- **Won’t Haves** - would like to have but probably will not have.

As the timebox progresses it is this prioritization of requirements that is applied. As the timebox is not extendable, the functionality of system must be trimmed in order to complete development work within the timebox to maintain continuous iterative development (Martin 1991). Utilising this approach it is the business requirements of a system that can be fully satisfied even if some of the operational requirements are not. The criteria is that it must be accepted as 'fit for purpose'.

### 3.5.4.3 Prototyping

Prototyping is essential to a RAD development approach, it is seen as a key characteristic (Martin 1991, Crinnon 1992, McConnell 1996, Beynon-Davies 1999). After initial requirements analysis through JAD workshops, timeboxing and descoping elements are applied, the resultant design/development is then prototyped to users. Maner (1997) illustrates this process in the following figure 3.14 below:
As shown the feedback is incorporated back into the design/development process through a number of iterations such that the eventual system is 'fit for purpose' meeting the needs of both the users and the business. This process is seen to represent a high quality system development. In intensive projects the process can take up to 21 days, but in more complex situations this timeframe is extended. The above diagram refers to focus groups for the feedback process, however this may also be associated with user groups who may or may not represent focus groups.

As the above figure illustrates a number of versions are achieved as the development processes evolve to supply additional functionality and incorporate business changes.

### 3.6 RAD and the DSDM Approach

More recently the DSDM Consortium have established 9 fundamental principles (1994) that they consider to constitute a RAD approach within the public domain (see table 3.1.) The aim was to provide a framework that combined the best
elements of existing methods with the practical experience of RAD developers. These 9 principles incorporate the essence of the original features and extend to address some of the evolving development issues that impact heavily on current systems development environments. Table 3.1 aligns the DSDM 9 principles against the original characteristics as defined by Martin. These are not prescriptive elements of a RAD approach but are adopted as appropriate to any development project.

<table>
<thead>
<tr>
<th>DSDM Principles 1994</th>
<th>Martin 1991</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Active user involvement</td>
<td>✓</td>
</tr>
<tr>
<td>2. Teams empowered to make decisions</td>
<td>✓</td>
</tr>
<tr>
<td>3. Frequent delivery of products</td>
<td>x</td>
</tr>
<tr>
<td>4. Fitness for business purpose</td>
<td>✓</td>
</tr>
<tr>
<td>5. Iterative and incremental delivery</td>
<td>✓</td>
</tr>
<tr>
<td>6. Changes are reversible</td>
<td>x</td>
</tr>
<tr>
<td>7. High-level requirements – base lined</td>
<td>✓</td>
</tr>
<tr>
<td>8. Integrated testing during lifecycle</td>
<td>✓</td>
</tr>
<tr>
<td>9. Stakeholders collaboration and co-operation</td>
<td>x</td>
</tr>
</tbody>
</table>

*Table 3.1 DSDM 9 RAD Principles & Martin’s RAD Philosophy*

Although the DSDM principles 3, 6, and 9 are not directly matched they are implied by Martin as incorporated within the RAD approach. In other words the frequent delivery of products is implied in the fast development iterations. Changes, and stakeholder co-operation are to some extent reflected through the integrated teams of developers and clients. Likewise the features defined by Martin not accounted for as a DSDM principle are just as he defined them, they are features and as such are regarded as integral to a DSDM approach i.e. Prototyping, JAD workshops and timeboxing elements of development.

The DSDM non-propriety RAD approach promotes a five phase lifecycle that consists of a Feasibility Study of the proposed project from both business and technical view points and the suitability of a RAD approach; a Business Study to define the high-level functional requirements and the key business areas involved;
a Functional Model Iteration that use prototypes to demonstrate the construction; a System Design and Build Iteration which aims to refine the functional prototype; and Implementation that deals with handover of system to users and a project review session. The DSDM consortium illustrate these phases in figure 3.15:

![DSDM Development Lifecycle](source www.DSDM.org)

Although only 5 phases are illustrated in figure 3.15 above there are also Pre-Project and Post-Project phases. The Pre-Project, Feasibility and Business Study phases are carried out sequentially, this is followed by iterative and incremental concurrent development of the three following phases until implementation, finally the Post-Project phase covers the activities of project audit and maintenance. During project initiation, business benefits are defined and prioritized according to the strategic needs of the business by the future end-users, using the developers to provide guidance on time, effort, and risk. Iterative development ensures that the resulting product fits the end-user's needs and incorporates the latest changes in the real-life requirements that cannot be anticipated at project initiation, real business benefit is therefore guaranteed (DSDM 2001).
3.7 RAD - A Genesis

As mentioned previously the iterative models of the 1980s addressed many of the former development problems, however literature puts forward usability concerns, low quality, high cost and lack of control as major issues for system development. Additionally the associated increase of user involvement impacting on time constraints, staff resources and scope creep presented new challenges (Martin 1991, Beynon-Davies 1996, Elliott 1997, Boehm 1999). All of these concerns questioned the potential of the contemporary fast development and delivery approaches to meet the increasingly uncertain and volatile nature of information system development environments, and their business needs.

RAD, originating from rapid prototyping approaches, was considered as a way of meeting the challenges presented by such uncertain and dynamic changing business milieux (Elliott 1997). First formalised by James Martin in 1991, it was adopted as a development lifecycle to help resolve the above problems (DSDM Consortium 2001). It was seen as a response to the commercial need for faster development than the more traditional methodologies could provide.

In the field of IS development approaches, RAD is considered to be fairly new. The nascent status of RAD as a development approach, and its original commercial emphasis has resulted in individual philosophies and perceptions of its rationale and application that have led to considerable debate about its suitability as a development approach for large complex Information Systems (IS). By the mid 1990s RAD was used as an umbrella term by different vendors and practitioners applying their own interpretations and approaches to encompass a number of methods, techniques and tools which, when integrated were thought to create a
synergy that facilitated rapid IS development (Martin 1991, Osborn 1995, Elliott 1997). In other words, the coalescence of different methods, tools and techniques working together would produce an effect greater than using the same components separately, many of which have existed since the 1960s and 1970s (Elliott 1997).

This unstructured evolution was partly due to the fact that RAD development was driven by the IT industry sector for commercial purposes to address the recognised need to deliver working business applications in shorter time scales and for less investment. As such it has had very little influence from the academic world (Beynon-Davies 2001). This spontaneous and extemporised evolution of RAD means that the rationale behind its use is not always clear. It is perceived as a development approach/framework, a development method for developers to change their development processes or as a set tools to improve development capabilities (Beynon-Davies 1999). An outcome of this ad hoc development of RAD meant that there was no agreed definition of a common RAD process.

Consequently, early RAD was seen as reactive, fragmented and sometimes considered as hype (DSDM 2001). Some refer to it as 1st generation RAD. As a result, in 1994 the DSDM (Dynamic Systems Development Method) was formed by vendor and user organizations with the objective of jointly developing and promoting a public domain RAD framework - a 2nd generation RAD. The intention was to combine the best elements of existing methods with the practical experience of RAD developers into a coherent framework. Additionally there was a recognised need in the market place for an Industry Standard RAD Framework, such that DSDM Consortium aimed to produce a 'de facto' development standard for building systems using an independent RAD framework. It focused on the principle of fixed timeframes and resources where requirements could be flexible.
within an iterative and incremental development approach. The DSDM approach was discussed in Section 3.6 above.

3.7.1 RAD – The Debate: Development Issues

As a systems development approach, RAD has both critics and supporters whose opinions, in some cases, are fundamental to individual philosophies and perceptions of its rationale and ad hoc evolution. Although RAD's nascent status and the lack of provenance is reflected by the limited availability of published academic material, there is however, substantial reporting of its application from the commercial sector. Thus RAD's origins can be placed more within a commercial development arena than an academic one. For example, on the academic front Beynon-Davies et al. (1999, 2000) report on 7 small/medium studies of RAD development projects, Jones and King (1998) who discuss 2 RAD implementations, and Girling et al. (1998) who present the benefits of a RAD case study. This can be compared to extensive coverage by practitioners reporting on small to medium commercial RAD projects that are too numerous to detail. Whilst RAD's successful application for small to medium development projects is not argued, and can be illustrated through an abundance of published articles, there is no similar body of literature reporting on, or analysing larger more complex development environments. This has led to considerable debate about its scalability across larger and more complex development arenas.

The literature raises three key areas that dominate the discussion, firstly the size, type and complexity of a project, secondly issues that coalesce under the umbrella of quality, and thirdly how an organization's culture can impact upon project development within large and complex project milieux. However, existing literature does little to clarify these issues thus establishing the need to extend the debate in

The literature does not present any definitive interpretation of project size in terms of small, medium or large, consequently the researcher accepts the categorization of project size imposed by the recognised authors. Likewise RAD’s suitability in terms of type of project and levels of complexity mentioned in the published material reflects an interest in whether the benefits of a RAD approach for small to medium interface projects can be achieved in relation to larger and more complex projects.


As previously stated RAD attracts both advocates and opponents, this part of the literature review turns its attention to the above issues through the examination and discussion of their published materials, in an effort to gain a balanced interpretation of the believed suitability or unsuitability of RAD as a system development approach across large, complex environments.
3.7.2 Project Size, Type and Complexity.

Even though RAD is becoming an increasingly accepted approach to IS development, the tenet is that RAD is not suitable for all types of project. RAD is generally associated with small/medium projects that are highly interactive, have a clearly defined user group and are not computationally complex. Consequently much of the existing literature tends to focus on small to medium size projects that does little to clarify the suitability of RAD across large, complex development arenas. Martin's (1991) original belief was that fast development of complex systems could be achieved if CASE tools were utilised and the project was decomposed into manageable modules.

This analysis was further examined through the beliefs of Avison and Shah (1997), Linthicum (1997), Beynon-Davies et al. (1998, 1999, 2000), and Howard (2002) who posit that there is a tendency to rule out using RAD for large scale projects, thus confirming that RAD is typically applied to relatively small low-key projects that are not computationally complex. RAD is seen as a good fit for simple application-development efforts with bite-size requirements, it is not considered suitable for complex Client/Server application developments. There is, however little explanation of the reasoning behind these views. An assumption is inferred that problematic areas experienced within small to medium sized projects will exponentially affect larger more complex environments, but as already established there are relatively few examples of such RAD applications to substantiate the conclusions drawn. Therefore the researcher believes there is a need to examine the various interpretations of RAD.
3.7.2.1 RAD – Framework, Method or Rapid Tools/Techniques?

Of initial concern are the different interpretations of RAD adopted that have resulted from the diversity of its ad hoc background. As previously explained it is perceived as an approach / framework, a development method, and/or a set of rapid development tools and techniques. It is often described in terms of its perceived benefits or problems experienced. Consequently, people's perspectives, in some cases, are fundamental to individual philosophies and perceptions of its applicability. In other words, people's views and opinions are influenced by their experience of its utilization. As a consequence there is a possibility that the views and opinions expressed as to RAD's suitability for large and complex arenas may be subjective, contributing to the beliefs held. Conversely, a response could be that the literature does not always fully communicate the interdependencies that exist between the utilization of the RAD approach, the emerging technologies, and advances in development methodologies, with regard to the increasing size and complexity of project environments.

Furthermore Bayer and Highsmith (1994) propose that there is some confusion between techniques (data flow modelling, entity relationship modelling), and the methods / methodologies that apply to them such as RAD via the DSDM approach. They suggest that within all projects data analysis is required and that different techniques are applied as appropriate, the goal is to accomplish the necessary analysis.

A number of prominent authors have reported on these very issues. For example Beynon-Davies stipulates that 'many people see RAD as a complete approach to IS development in that it covers the complete lifecycle, from initiation to delivery'
(1998, p2). This view is supported by the DSDM Consortium (2001) and Osborn (1995) who conclude that RAD is more of a framework than a method and not always suitable where requirements are fixed at a low level. Likewise Jones and King (1998) suggest that a DSDM RAD approach does not prescribe tools and techniques but instead serves as a high-level project management framework.

However Beynon-Davies (1998) continues by emphasising that it is not an IS method in itself as it adopts a toolkit or contingent approach to systems building rather than a prescriptive one. His opinion of its suitability is that RAD, as a development process, depends on the type and size of system, the project management approach and level of management commitment, end-user involvement and the ability of the team to make fast authoritative decisions. At this point in his research Beynon-Davies suggests that RAD is particularly applicable to small and medium scale projects but that large projects should only be considered if they are split up into smaller components each of which can be delivered independently, this aligns to Martin’s (1991) original views. However, in 2000 he advises that RAD is not suitable for all types of project and goes on to question its appropriateness within IS development practice. He reports that developers continue to reiterate their concerns over issues of cost, scalability, benefits and culture in relation to RAD.

Conversely Osborn (1995) proposes that RAD as a framework of tools and techniques is appropriate for developing systems to support high-level unstructured processes suggesting the computerization of large transaction systems as an example. He believes that its strength lies in developing interface oriented systems that already have a data infrastructure in place, but recognises that its practical success depends on the quality of existing system architecture.
and information infrastructure. However Linthicum (1997) states that there is a tendency to focus on the technology rather than the business. He believes that it is the system requirements and not the RAD process that should drive the systems architecture and design. Linthicum further proposes that a drawback of the iterative nature of RAD is that it is perceived as reducing the need for specifying business requirements, and thus some developers may not see the need to apply a rigorous application development lifecycle. He surmises that RAD is only one component of the application development lifecycle, it is not the lifecycle itself.

Bowen (2001) presents the criticism that where a system is generated using RAD tools, its implementation may not be optimised and consequently scalability across larger systems is an issue. He infers that using RAD tools does not equate to using a RAD development process. Whilst Goa (1998) reflects that although the poor performance of early rapid prototyping made them unsuitable for large or complex systems, the evolution of more sophisticated tools using standard software development techniques is thought to make it more appropriate for large systems development.

RAD is also criticised in terms of its application for certain types of projects. One consensus is that RAD is not appropriate for real-time or safety critical systems, very large infrastructure systems, computationally complex systems or where functional requirements have to be fully specified before any program is written (Beynon Davies 1998, DSDM 2001, Raffoni 2000). A further view is that RAD tends to fail when applications must inter-operate with existing programs or where the system cannot be modularised for development or it is not possible to utilise 4GLs. Additionally, RAD is not considered suitable for projects where new ‘bleeding edge’ technology is introduced because of the associated risks of the
complexities involved. It is associated with a lack of rigor that results in fragile systems that do not scale well particularly where extensive distribution is concerned. For example in horizontal or mass markets because the use of prototypes tends to raise user expectation to unrealistic levels (Linthicum 1997, Maner 1997 and Cross 2003).

In addition some developers see RAD as a collection of practices rather than a single approach because they believe its origins are based within database centred systems, and therefore draw a conclusion that that it does not apply to any kind of system development. Others associate RAD with faster development rather than as a meaningful methodology that can be delivered as a loosely integrated, but not prescriptive, series of ‘best practices’ (McConnell 1996).

All of the above views and opinions reaffirm that it is the fact that RAD is seen in the light of its application that influences the individual perceptions held. However, the diversity of beliefs put forward highlight a central issue – that of quality or a perceived lack of quality in relation to RAD. Thus the debate continues.

It should be noted that ‘RAD’ as a development term is more commonly expressed through the DSDM approach which is described as the de facto standard for rapid application development in the UK (Barrow and Mayhew 2000). Since that time authors are reporting on Adaptive Software Development and Agile Development which bear a family resemblance to RAD.

The researcher has discussed below some of the current arguments surrounding the quality issue. It is anticipated that the findings from the analysis of the research case study will effectively contribute to this discussion.
3.7.3 RAD – Quality Systems Development or Quick and Dirty Solution?

In 1991 Martin originally defined RAD as ‘development lifecycle designed to give much faster development and high quality results than the traditional lifecycle’ (p2). Fast development, achieved through the use of CASE tools, was associated with lower cost due to shorter development schedules.

He did not support the view that speed was inversely proportional to quality, but contrarily stated that ‘there should not be a compromise between speed of development and quality’ (p3). He maintains that high quality, lower cost and rapid development ‘go hand in hand’ (p5) providing that the methodology adopts the appropriate tools and techniques.

More recently Maner (1997) extends this theme by questioning the lack of compromise between speed, quality and cost. He believes that there is some negotiation required between these three factors. He maintains that it is the tradeoffs between schedules, economy and product quality that can be aligned to the successful use of RAD. He concludes that compromise between either economy or quality produces a fair chance of success but if both economy and quality are compromised then a greater chance of success is achieved. However, he qualifies this by explaining that negotiating quality does not mean accepting a higher defect rate. Quality is therefore not abandoned but redefined as ‘fit for purpose’, a principle supported by the DSDM framework, rather than representing a technical excellence and zero defects. This means that flexibility and goal orientation become more important than the technical expertise/precision that is associated with traditional methodologies. He clarifies further by explaining RAD facilitates an early representation of the system that is both acceptable to the
client, and feasible for developers. The associated iterative nature and incremental delivery limits exposure to the forces of change, and saves development time although this may be at the expense of economy or quality.

Boehm (1999) on the other hand puts forward the view that the trend of reducing the schedule of a software development project is becoming more important than reducing its cost. This is borne out by the different RAD models that present effective methods for reducing cycle time. He puts forward the examples of GRAD (GeneratorRAD), CRAD (CompositionRAD), DRAD (DumbRAD) and FRAD (Full system RAD) each of which reflects a component of the development lifecycle. However quality is not a single element, it is not achieved through any one factor, but rather through a combination of factors. It is by involving groups of developers and users working together in small teams throughout the project lifecycle, the use of CASE tools and prototyping, and by emphasising the importance of reusability of design, procedures and components that can ensure standards of quality (Martin *Ibid*⁴). It is the increased involvement of users during the design and development stages, together with prototyping that visualises the evolving system that results in a product that not only meets the users needs, but also occasions lower maintenance costs. He puts forward figure 3.16 to illustrate this view:

---

⁴ The same book by this author previously referenced.
The shift away from the traditional monolithic linear approach to system development towards evolutionary and iterative development has engendered a broader interpretation of system quality. Bayer and Highsmith (1994) posit that the literature focuses on particular development techniques such as JAD, prototyping and timeboxes. They put forward the argument that for success within larger organizations there is an increased necessity to focus on the customer and the product. It has become aligned to clients’ perspectives and is an intrinsic part of a system’s success. Whereas traditionally success was assessed in terms of the system being delivered on time and in budget, more recently it is the ability of the system to meet the ‘fit for business purpose’ principle. This is in line with the move towards increased and sustained levels of user involvement throughout the project. The ability to meet users and business requirements are seen as integral quality factors. Supporters of RAD argue that RAD has these inherent quality principles because the users become an integral part of a RAD approach. Therefore systems tend to meet user expectations thereby reducing the amount of maintenance costs.

Additionally the benefits of using CASE tools are manifest in increased productivity such that system specifications are more accurate, more complete and have a higher consistency. CASE tools used as part of an iterative and evolutionary RAD process ensure that systems are accepted and meet requirements (Avison and Shah 1997). The essence of evolutionary development is change, the ability to incorporate business changes as the project progresses adds value to the system that in turn engenders higher levels of quality.

An alternative view to quality is presented by Demarco and Lister (1987), who suggest that quality need not necessarily be factored in, it is the people who produce the quality, they need to be part of a culture that is quality oriented. Quality and productivity levels fall where there is little job satisfaction, motivation is low and employees are dissatisfied. They further state that the ‘cult of quality is the strongest catalyst for team formation’ (p151). They maintain that it is the motivational need for satisfaction that drives the impetus for quality with a team environment. However, unrealistically ambitious software schedules put people under pressure and this reduces both productivity and motivation, and consequently quality. They state ‘people under time pressure don’t work better; they just work faster’ (p18), the message is that in order to work faster, both job satisfaction and quality are sacrificed.

This point is additionally emphasized by the DSDM consortium (2001) who maintain that the collaboration of stakeholders in JAD workshops together with the review cycles produces more focused information gathering activities where
process redesign is led by the user community responding to the need for faster system development. It represents the shift away from the previous concept of the developers working for the clients, to one of them working with the future end-users to achieve the RAD's goals of speed and quality. The DSDM associate return on investment (ROI) in terms of time as an essential criterion for the project success. They stipulate that the iterative development process results in a product that fits the end-user's needs and incorporates the latest changes in the real-life requirements – thus real business benefit is therefore guaranteed (DSDM 2001).

Contrarily, although concurring with views that promote speed of development and rapid adaptability as project goals, Alavi (1984) suggests that time and cost savings result in lower overall system quality, that is vulnerable to an inconsistency of business or corporate standards. Martin advises that bureaucratic environment should be avoided because 'bureaucracy is the enemy of speed' (1991, p128). He stresses that the formal hierarchical procedures inherent in a bureaucratic environment prevent rapid development.

As the reader is beginning to observe there is a dichotomy of viewpoints regarding the interpretation of RAD in terms of the quality of the product developed. Critics of RAD have attached a ‘QRAD’ label to it, calling it a ‘quick and dirty application development’ (Merkert 1996, Linthicum 1997, Raffoni 2000). Other meanings for the RAD acronym are Rapidly Achieving Disaster, Really Awful Development (Saleh 2001), Really Awful Design (Day and Watson 1999), Rapidly Achieving Disaster (Day and Watson 1999). However supporters of RAD argue that RAD has inherent quality principles and can be applied to any software project depending on the approach taken (Graham 1989, Osborn 1995, Elliott 1997, Maner 1997,
Howard (2002). It is interesting to look at some examples of these viewpoints in more detail.

Some developers have understood RAD to mean incrementally speeding up each stage of the traditional software cycle; others interpret it as automation with extensive use of software development tools. As mentioned above, Howard (2002) reports that many believe RAD stands for ‘Rough and Dirty’ development suited only for small, relatively low key projects. He suggests that RAD can be an excuse for sidetracking the disciplines of software engineering standards thereby reducing quality for faster systems delivery.

He continues that because it is a common belief that speed and quality in software development are incompatible, RAD is accused of being anti-quality, you can have one or the other but not both at the same time. This view is supported by Raffoni (2000) who purports that the speed of development can be aligned to the truism that faster development can mean faster mistakes, in other words get it wrong faster.

Linthicum (1997) refers to RAD as ‘The Good, the RAD and the Ugly’. He believes that this is due to its perception as a ‘silver bullet’ for application design and development that reduces the need for specifying business requirements and system design. The iterative approach is thought to reduce the need to apply a rigorous application development lifecycle.

In contrast to the above opinions expressed, others argue that depending on the approach adopted, RAD and its techniques can be applied to any software development project. There are a number of authors who support Martin’s original
view that ‘fast development does not mean ‘quick and dirty’’ (1991 p3). Similarly McConnell (1996) argues that RAD is not a quick and dirty way to produce mediocre systems, and that it is not restricted to small new developments. He believes that quality is achieved through the use of small collaborative teams, increased user involvement and effective communications. Although RAD is criticised for lacking rigour and is associated with developing fragile systems that lack scalability, a more disciplined RAD approach containing well defined and understood processes, and getting the right people is perceived as the way forward.

This argument is further supported by a PEP\(^5\) paper (1997) that reports RAD is not a ‘cookbook approach’ to systems development, neither is it restricted to small new developments. RAD focuses on the rapid development of critical system functionality. It combats scope and requirements creep by limiting the cost of change through iterative development and early user feedback, thus adding value to the development process that contributes to quality. Cross (1998) put forward the view that a disciplined RAD approach would become the \textit{de facto} approach for the development of software intensive systems.

Thus the literature expresses some controversy about the quality issues with this genre of development approach. Supporters argue that there are inherent quality principles. They see the ‘fitness for purpose’ principle, less waste and high user involvement to get the design right and to meet customer requirements as integral quality factors. Others argue that descoping business needs reduces the quality and the capability of the product to meet all user needs. It revolves around the ability of the people involved to identify those requirements that enable a system to

\(^5\) Computer Science Corporation paper 1997.
satisfy both business and user needs. Management of requirements elicitation, user involvement and user expectation are all interrelated, they are linked through the use of timeboxing, a critical factor for a RAD-type approach. Timeboxing is linked to both management of, and levels of user expectation. The premise is that if users are involved in JAD sessions during design and development activities they have an awareness of the system being developed and hence more realistic expectations.

In conclusion the issues of quality and speed have been raised, and the two alternative positions expressed in the literature have been discussed. The dichotomy that exists between those who regard RAD as a poor quality approach, and those who believe to the contrary is reflected in the debate of whether quality and speed are mutually exclusive. In other words whether increasing speed through the use of development tools will necessarily reduce or increase quality; and whether increasing development speed by scoping functionality produces poorer or better quality systems. Some see quality in terms of software and technical excellence; others perceive it as system functionality and usability. This raises the question - is quality embedded in the actual quality of the software of the system being developed or in the quality of its actual use? It is possible to produce a quality system from an engineering standpoint that never gets used. In practice quality is a relative term; it depends on individual interpretations and beliefs, and it is a balancing act, a trade off between the factors involved.

3.7.4 Implications of Culture on a RAD Development Process.

Success of a system development approach is tempered by the absolute nature of an organization i.e. its context, and its limitations. A methodology does not necessarily map directly onto an understanding of the organization, its rationality
or the context of its users (Coughlan and Macredie 2002). Therefore where there is a cultural mismatch, the benefits of the approach chosen will be either lost or unrealised. Where organizational cultures have evolved it does not follow that system development cultures have evolved along the same lines. It is necessary for the development process to undergo a transformation from a traditional static and documentation driven process towards a dynamic, evolutionary and product orientation (Bayer and Highsmith 1994). This view is extended by Highsmith (2000, 2002) who accentuates the need for systems development to respond to the changing culture of development arenas. He refers to the increasingly dynamic and adaptive nature of current software project management where technological and business changes accelerate the failure of static management practices. He stresses the cultural dichotomy of the traditional optimal approach versus the iterative and adaptive methods that emphasize change, improvisation and innovation.

3.7.4.1 RAD and Cultural Changes

Although there is very little reporting of RAD within large complex development projects, literature indicates that for such environments a successful RAD approach necessitates management, cultural and human changes. One such proponent of this is McConnell (1996) who puts forward cultural and management changes as the two most problematic barriers. RAD projects require cultural elements that are different from those in more traditional applications development because RAD requires people to behave in a different way. He suggests that without radical shifts in organizational attitudes and structures, and the mindsets of people, many projects failed because the change to new methodologies, methods and techniques did not fit within the organizational culture. He further proposes
that the cultural change that is required not only reflects a joint development ethos but also identification of accountability within team oriented work environments.

More specifically RAD enforces changes in a RAD-type project's structure that is different from the more traditional development arena i.e. the management approach. It necessitates a higher level of management commitment, increased and sustained user involvement, and particularly the ability of the development teams to make the fast authoritative decisions that facilitate the rapid development ethos. These factors are seen as crucial and their emphasis is considered greater across the larger, and more complex projects (McConnell 1996, Beynon-Davies 1998, DSDM 2001). However there are different elements to consider when examining culture, these are discussed next.

### 3.7.4.2 Culture and Stakeholders

Integral to success of a system development project within RAD is the view that the success is influenced by the users involved. As stakeholders they have a significant influence on the success of the project (DSDM). In fact one of the main principles of the DSDM approach is the need for co-operation and collaboration of all stakeholders. Thus management of the people concerned is seen as a key factor.

Much of the literature reporting on the cultural dimension concentrates on the context of the organization. However the human issues of people's attitudes and behaviour are influenced by the ‘format’ of the organization. Thus the researcher acknowledges that issues surrounding the impact that culture has upon a project development extends beyond the limited remit of this Section. Therefore rather than present a fragmented approach within this section, a more complete and
deeper analysis is required, consequently this area is explored further in terms of organizational, group and personal culture in Chapter 6.

3.7.4.3 Blame Culture

Within large and complex development environments where the organizational nature is one of bureaucracy and hierarchical status, a blame culture is particularly prevalent. A blame culture is counter-productive to development approaches such as RAD because the inherent nature of line management driven responsibility inhibits the decision-making processes. RAD necessitates a collaborative working culture that is dominant. Therefore as a development approach it needs to promote an anti-blame culture, otherwise problems associated with a blame culture will become manifest within the project framework. Allocation of responsibility should be removed from the individual status to be accepted within a team culture - it is a question of joint responsibility. An IS approach must work for an organization, otherwise little benefit is provided (DSDM 2001).

3.7.4.4 Team Culture

RAD promotes a strong emphasis on team culture within system development environments. Within large organizations it necessitates moving away from a hierarchical, top-down organizational structure to an empowered, team-oriented development environment (Bayer and Highsmith1994). There is a link here to user involvement and the formation of development teams. Carmel (1995) believes that RAD projects can fail during implementation not only because the mixed teams were incorrectly selected but also because of the lack of strong people management.
There is considerable difference between a real team and a group of people who work together. A jelled team is a group of people so strongly knit that the whole is greater than the sum of its parts, but which can just as easily degenerate into being less than the sum of its parts (Coughlan and Macredie 2002). Where an organizational culture stresses individuality and individual accountability i.e. a blame culture, the concept of team culture is more problematic.

The PEP 25 (ibid.) paper suggests that there are three levels of change required within a large organization when applying a RAD development approach. Firstly there is a need to give people the time to adjust to the physical changes, such as the project environment, development tools and JAD workshop scenarios. Secondly there is an intellectual change that refers to the understanding of the new approach to development, to understand the different cultural beliefs and values. Thirdly an emotional change is needed to engender the trust and cooperation for collaborative working between the business and the developers.

Mayhew et al. (1998) extend this issue of change to incremental prototyping and put forward the framework that Hirschheim et al. (1996) presented which suggests there are three classes of change relevant to IS development. These are cosmetic changes that represent minor amendments, local changes that concern a given sub-system that require some prioritisation and strategy, and global changes that impact across the development project and which require authorisation and version control.

Consensus drawn from previous IS research has shown that change must be managed effectively (McConnell 1996, Jones and King 1998). Martin (1991) suggests that cultural factors can lead to system failures and recommends that

3.8 Project Factors of RAD Development

One of the aims of this chapter is to create a 'Table' from the analysis of existing literature that represents the key factors affecting the application of RAD. The researcher's intention is to extend that table with analysis drawn from the research case study of an actual large, complex RAD-type project, and to present a model of critical success factors that may influence and impact upon other RAD-type project approaches within similar development milieux.

To achieve this a context must be drawn from existing beliefs that are fundamental to RAD. As can be seen from the plethora of views and opinions previously discussed, this task was problematic due to the extensive range, and diversity of terms that have been used to describe RAD, its problems and benefits. For example, in 1991 Martin originally described RAD in terms of four essential aspects (tools, methodology, people, and management); he characterised the main features of RAD as user involvement, prototyping, CASE tools, SWAT teams, JAD workshops and timeboxed development, and defined RAD's philosophy as high speed, high quality and lower costs. The DSDM Consortium put forward its 9 principles (see table 3.1, p94) as the fundamentals of a RAD approach. Beynon-Davies (2002) refers to RAD's features (project type, project team, product and development process). The literature, on the other hand,
discusses RAD in terms of issues, components or elements associated with success or failure. Thus any direct comparison is difficult. However, it has been possible through further evaluation to identify the common ‘threads’ that are woven throughout the literature. Consequently an initial ‘Table’ of factors has been produced that represents the main issues of a RAD-type approach from a literary standpoint. This is set out in table 3.2 below:

<table>
<thead>
<tr>
<th>PROJECT FACTORS</th>
<th>HUMAN FACTORS</th>
<th>ORGANIZATIONAL FACTORS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of Project</td>
<td>User Involvement</td>
<td>Culture</td>
</tr>
<tr>
<td>- intensive</td>
<td>- active / sustained</td>
<td>- public / private</td>
</tr>
<tr>
<td>- non-intensive</td>
<td>- committed</td>
<td>- nature of structure</td>
</tr>
<tr>
<td>Type of System</td>
<td>Decision-makers</td>
<td>Management /Commitment</td>
</tr>
<tr>
<td>- low complexity</td>
<td>- empowered</td>
<td>- top management commitment</td>
</tr>
<tr>
<td>- high complexity</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Size of System</td>
<td>Behaviour</td>
<td></td>
</tr>
<tr>
<td>- small</td>
<td>- change receptive</td>
<td></td>
</tr>
<tr>
<td>- medium</td>
<td>- team working ethic</td>
<td></td>
</tr>
<tr>
<td>- large</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Table 3.2 A Table of Key Issues Affecting RAD Developments*

The three groups of development factors, in table 3.2 above, are areas that are external to the RAD development process but which are important to the receptivity of a RAD approach. This research study has taken the DSDM model as a given, and it is these factors that are a key emphasis to the research project because it explores the relationships between them, and their affect upon the RAD-type approach adopted. The assumption was that a toolkit of RAD techniques such as JAD workshops, timeboxing, incremental development and so forth, which are the factors internal to RAD, would be used in the development approach.
As illustrated in the above table 3.2 the project factors are those concerning type, size and complexity of system being developed, the organizational factors are those pertaining to culture and management, and the human issues are the users, their involvement, behaviour and decision-making abilities. It is the human factors that present the most debate. Consequently it is these stakeholder groups that have been explored through this research case study, and it will be interesting to compare this table against the analysis and findings of the case study to identify areas that can be added to, extended, or removed and thus create a model of critical success factors for future RAD development projects operating within a similar development arena.

3.9 Summary

Advancements in technology and the increasing sophistication of development environments have facilitated a maturing of development methodologies towards architectural frameworks able to support iterative and incremental development such as RAD.

Initial interpretations of RAD were threefold, and as such its rationality has been tied to individual experiences of its application. However consensus lies in the understanding of its ability to accommodate system development within dynamic and volatile project arenas, and there is some accord amongst protagonists that RAD is best suited to small to medium sized projects with low levels of complexity. For larger and more complex projects special consideration needs to be given to management and control aspects where the inherent formality of organizational structures (in particular bureaucratic cultures), procedures and communications hinder the RAD philosophy of collaborative and co-operation working and the ‘fit for purpose’ ethos.
Literature acknowledges the benefits and problems associated with a RAD-type development approach. It suggests that failure to adequately understand both the organizational and development environment cultures can lead to project failure. The areas of discussion put forward by the IS community reflect the issues of most concern. For example, factors attributed to project, organizational, and human aspects reflect a need for cultural acclimatization or change. More specifically it is the issues of project type and size, management commitment, degree and sustainability of user involvement, decision-making skills and stakeholder behaviour that raise the most debate. These problematic areas, which present the most challenge, are increased when the additional factor of technological innovation integral to the project is added.

Additionally it is the Gestalt concept that is important. RAD as an organized whole is greater than the sum of its parts i.e. tools and techniques. Simply combining JAD, toolsets and ambitious timeframes does not make a RAD environment. Adopting a RAD-type approach is more complex than just coercing people into meeting unrealistic goals and applying RAD techniques and tools.

However success may be difficult to repeat because no two projects are the same or evolve in the same manner. In synopsis RAD’s suitability is regarded as dependent on the type, size and complexity of domain or application, and the associated organizational contexts of structure, culture and people factors. It is these issues that are further explored in Chapter 4.
CHAPTER FOUR RAD Development Factors and Stakeholder Issues

4. Introduction

This chapter considers further the stakeholder issues. It explores in greater detail the areas that are external to the RAD development process but which are important to the receptivity of a RAD approach (see table 3.2, p121). The aim was for the researcher to gain a sufficient depth of understanding of these areas such that analysis of material gathered would contribute to the fulfilment of research questions 3 and 4:

Research Question 3: What are the major influencing factors that affected the development process of the research case study project?

Research Question 4: Were the problems experienced attributable to the RAD-type development approach adopted, or to other aspects of research case study?

The literature review was two pronged. Prior to commencing fieldwork within the project development arena, an initial in-depth and comprehensive study of the current status of RAD as a development approach was carried out. Subsequently an on-going systematic analysis of the literature within the IS domain driven by the issues identified through research case study was conducted. For instance, the stakeholder issues of user involvement, requirement elicitation, decision-making, user expectation, and communications were identified as affecting the development project. All experienced problems and required further examination within the literature rather than just from a pure RAD aspect. Furthermore, in this research case study, culture was a major influencing factor and rather than present
a fragmented approach within this section it is the subject of a further chapter (see Chapter 6).

As a consequence the researcher was able to gather the data from the development arena without bias or influence from the literature, which may have occurred if all areas were explored prior to fieldwork. In this way the researcher was not looking for specific or identified causal relationships but rather concentrated on understanding and evaluating the issues that were indigenous, and therefore significant to this case study environment. The aim was then to reflect upon the opinions put forward in the literature in conjunction with the data analysed from which to draw conclusions. Such conclusions would then inform the current debate concerning the utility of using a RAD-type development approach for large and complex IS systems development. Consequently this chapter details the above issues as explored in the nature of a literature review.

4.1 User Involvement

Martin states that during the design phase 'end-users participate strongly in the non-technical design of the system' (1991, p81).

User involvement in systems development is a complex issue. It is multi-faceted. There is much discussion surrounding the definition of a user, the identification of the right users, the degree of their involvement and levels of availability and commitment. Traditionally user involvement is linked to user expectation and user acceptance, both of which affect the success of a system. User involvement is not confined to these areas but also had a part to play across requirements elicitation, decision-making, and communication issues that are explored accordingly.
4.1.1 User Definition

In order to discuss user involvement it is first necessary to understand what is meant by a user. *The term 'user' is a somewhat confusing one* (Beynon-Davies 2002, p183), different people apply different interpretations to the term. It is not possible to provide an unambiguous definition of 'user' (Carmel 1993). It has been used to define the non IS people, the non-technical people, and also the operational people of an organization who are affected by the system being developed. Users are considered to be one group of the stakeholders of the system – the differential is the degree to which they are involved. More recently the term 'stakeholder' is often used to emphasis that some users are not end-users but are the more indirect users/stakeholders, for example senior managers, business managers, customers and so on. The term end-user refers to those people who actively work on the system on a regular basis. Willcocks and Mason (1987) concur with the view that stakeholders are those people who will be significantly affected by the system or have a material interest in it. Their involvement is considered essential for successful development (Beynon-Davies 2002). Stakeholders can be both internal and external to the system being developed. However within the literature cited authors have predominately referred to 'users' collectively rather than to distinguish between stakeholders and end-users. The researcher acknowledges the distinction that has been made between the two terms but maintains the usage applied by the authors sourced to maintain the integrity of the literature cited.

4.1.2 User Involvement and User Participation

User involvement and user participation are both terms used to describe taking part in and/or contributing to system development.
Some literature refers to user involvement and user participation as being interchangeable in relation to IS development. However some studies draw a distinction between them. Hirscheim (1983) for example, maintains that all projects have some level of user involvement but users are not necessarily participating in the content development processes. He maintains that participatory design (PD) is not the same as user involvement. The difference lies in the relationship between the stakeholders and the developers in the project development. PD extends to more than just user input, it involves decision-making and content analysis rather than a mere statement of needs. Additionally PD attends also to the social elements in the same vein as Checkland’s Soft Systems Methodologies, not just the technical issues. Mumford’s ETHICS (1983) approach is an example of a development style that leans towards social design.

Barki and Hartwick (1989, 1994) believe that user involvement relates to the importance, relevance and understanding of the system to the stakeholders i.e. their attitudes towards the system. Whilst user participation, they believe, defines a more active involvement during the design, development and implementation of the system in question i.e. decision-making. The latter also incorporates some form of responsibility. There is an association between end-users being involved / participating and increased user satisfaction and acceptance. Kappelman and Maclean (1991) however have combined these terms together into ‘user engagement’ that encompasses both interpretations i.e. the attitudes of user involved, and the activities of user participation to reflect a holistic user relationship between the IS and the development. Baronas and Louis (1988, cited in Kappelman and Maclean 1991) believe that involving users in decision-making activities particularly during implementation increases their interest in the system providing them with a sense of control, which in turn augments user satisfaction,
user acceptance and system success (1988, p114). It is this view that is most commonly supported in the literature.

4.1.3 Right Users

Consensus in the literature emphasizes the importance of involving stakeholders in system development projects. Views expressed by Martin (1991), Hirschheim (1983) and Beynon-Davies et al. (1996, 1998), support the view that for large and complex development environments, where it is not practicable to involve all system users, the issue of identifying the right users/end-users is crucial to system development. If the identification and allocation of the users is incorrect, particularly in requirements elicitation, then the systems scope will be inaccurate. Typically user involvement has referred to the end-users of the system, such that if these needs are met then they will accept and use the system. Thus the link is made between user involvement and success of IS development projects.

Beynon-Davies (1996) however raises the issue that the right users may not necessarily be the end-users but the middle managers; the norm is that they are generally selected from the ‘pool’ of potential users. He discusses the hierarchy of users relative to a project; each has different levels of knowledge and different degrees of availability and commitment. Their characteristics are that they are representative of the entire user community, they are enabled to make authoritative decisions, and that they possess a positive commitment to the development project. However those involved may have differing levels of empowerment and consequently differing levels of authority over change requests. Those with limited empowerment will be confined to superficial changes, whilst others will be empowered to make more serious changes. There is a link between the level of empowerment and the ability to demand changes to the development
Moreover it is not always the end-users who possess the characteristics required i.e. decision-making qualities, or who are the holders of key business knowledge. Additionally the end-users traditionally are those maintaining the day-to-day business activities and thus their availability is limited, consequently commitment is only partial. (Ljubic and Stefancic 1994, Barrow and Mayhew 2000). The suggestion is that if the right users are involved, the right system will be successfully developed.

4.1.4 User Expectation

In the RAD literature there is a consensus that user expectations are linked to levels of user involvement. If user involvement is low or limited, or the right users are not correctly identified, there is a higher risk that their needs will not be met. As a consequence user expectations will not be in line with the system being developed and difficulties will be experienced with user acceptance. However, one view is that sustained involvement of users is costly, and conflicts with maintaining the day-to-day business activities. There is some suggestion that it is less effective within larger project environments as it proves difficult to facilitate the meetings necessary to reach consensus on decision-making. Additionally, some authors have associated the iterative development nature of a RAD-type development approach with a reduced need for specifying business requirements. This is because each development iteration amends the deficiencies of its predecessor (Martin 1991, McConnell 1996, Linthicum 1997). Contrarily, as previously mentioned, critics of RAD maintain that reduced requirements gathering leads to lower levels of user satisfaction and user acceptance, thus reducing the quality of the system being developed.
However, alongside these issues the researcher acknowledges that effectiveness of user involvement is also associated with user expectation and should be discussed. Willcocks and Mason (1987) raise the question of the appropriateness of user participation. Which users should be involved? What degree of influence should they have over decision-making and change requests? These are particularly interesting points as they relate significantly to the case study since they impacted upon the development process ultimately affecting delivery schedules.

Carnall (2003) reflects on the importance of engaging key stakeholders during periods of change, in this context the researcher is referring to the user community⁶. For user acceptance to be successful valid knowledge is essential in order to manage the uncertainty that people feel. Feelings of uncertainty potentially promote negative views of a system being developed. Learning is a key issue in the need to build new mental models of the changing business processes. Without the learning there is insufficient knowledge to engender acceptance and a sense of loyalty onto the developing system.

Markus (2004), in her recent paper about Technochange, advocates that when organizational change involving IT is concerned, it is necessary to focus on the people who are affected by the change. Martin believes that many organizations under estimate the importance of training, he maintains that end-users must be trained, that it is a vital element. He states: ‘good user training is essential for achieving success with a new system’ (1991, p329). He also recognises that training should be developed in parallel with system development. This concern is shared by Clegg and Barker (1994) who suggest that high-levels of user

---

⁶ user community refers to both direct and indirect users within the research case study.
involvement equate to reduced training needs, as those participating will be familiar with the evolving system.

4.1.5 Disadvantages of User Involvement

There are also other risks associated with user involvement, for example the tendency of end-users to request new requirements during design and development that result in feature creep, stretching already tight schedules and increasing the scope of the project (McConnell 1996). Traditionally creeping scope was managed through up-front identification of requirements early in the project but where an iterative approach is applied within a volatile business environment it is inevitable. It is the ability for change to occur that provides business value. However user changes should be both realistic and reasonable and managed rigorously to avoid missing development targets (Highsmith 2002).

4.2 Requirements Elicitation

Requirements elicitation is crucial to all development projects. It is the requirements that provide the basis for the design, development and implementation of the system being developed. Requirements elicitation is linked to user involvement, as discussed above. Their involvement in the design and development processes is accepted as a more effective way of understanding user requirements. This is compared to the one-to-one interviews/meetings of early requirements gathering favoured by the more traditional linear development approach. Typically JAD workshops are the most common requirements gathering technique associated with a RAD-type development approach.
4.2.1 Issues of Requirements Elicitation

Easterbrook (1991) acknowledges that there is always a degree of uncertainty in the original requirements identified and that it is necessary to ensure that all perspectives are adequately represented. Cross (2002) claims that the reasons for failure in software development are largely the same as they were 30 years ago. The most frequently cited reason for failure was poor management of requirements characterised by repeated changing of requirements, requirements that are not well understood and, requirements proliferation. He states that valid criticisms of an undisciplined RAD approach are lack of rigour, leading to systems that do not scale well and which raise end-users and management expectations to unrealistic levels.

A survey carried by The British Computer Society (2001) recognised the difficulty involved in defining requirements. It states that poor management of requirements and project scope are common causes of failure. The Survey reveals that across all sectors published, only 13% of IT projects were successful, but more significantly for development projects this figure is less than 1%, i.e. delivered on time, in budget and to specification. It cites change requests resulting from both internal and external factors as significant factors. In the opinion of three-quarters of IT project managers, the Survey reports, no project had ever been delivered to the initial specifications. However, this may reflect the dynamic nature of current business environments rather than imply ineffective requirements gathering.

The general consensus expressed in the literature is that requirements elicitation, although an open-ended activity, is of paramount importance such that the resulting system is not only 'fit for purpose' but actually meets the dual objectives
of user and business needs at implementation. Eva (2001) argues that traditionally requirements elicitation ‘presupposes a tidy and unambiguous definition of requirements, which incorporates solutions to problems with the operations of the current system’ (2001, p102). This suggests that a fixed set of requirements could be identified and gathered. However he points out that in practice this is not the case, system requirements are not a discrete entity. If it is the purpose of the system to provide information support rather than to purely automate business processes then a discrete requirements specification is not sufficient. In reality interaction between the analysts and a multiplicity of system users is needed to provide a broader perspective of business activities and the information needed to support them.

Although highly error prone, the move towards a more collaborative data gathering approach that encompasses both user context and organizational factors increases the degree of success. With regard to large and complex systems, the more complete the requirements the more successful the system is in terms of user expectation and user acceptance (Coughlan and Macredie 2002). RAD utilises Joint Application Development (JAD) workshops to achieve the necessary level of requirements to meet the ‘fit for purpose’ concept. Involving users in the JAD design and development processes is accepted as a quicker and more effective way of understanding user requirements in contrast to the one-to-one interviews/meetings favoured by the traditional development approach.

4.3 Decision-Making

The RAD-type approach relies heavily on the ability of those concerned to make empowered decisions in a timely fashion, without having to resort to higher management for guidance or control. Decision-making is closely linked to the
issues discussed in user involvement, and encapsulates the concept of 'fit for
purpose' of the system to meet core business needs, and involves consensus
about development decisions. As already discussed, application development has
moved away from 'starting all over again' to 'transforming what already exists'. It is
the development process that has centre stage not the product. Systems are no
longer built to last, they are built to change (Smith and Fingar 2002). RAD means
'fit for purpose' rather than the former interpretation of traditional methods as 'zero
defects' at the expense of business need.

Morgan's views raise the issue of empowerment. Decision-making is inhibited by
hierarchical cultures, he states 'the limits of empowerment are usually quickly felt
as people run into the constraints imposed by the existing hierarchy' (1997, p169).
The nature of decision-making is culturally rooted. Who is involved, who makes
the final decision and where the decisions are made, are reflections of the
organizational culture (Schneider and Barsoux 1997). Participants need to be able
to make fast, authoritative decisions in order to progress development activities.
Where there is a culture of line management decision-making this can prove
difficult for people who are used to deferring decisions up the management line.
Empowerment focuses on people rather than the process. The emphasis is on
who makes the decision rather than the input into making the correct decision and
avoiding blame. However it is often the case that development teams are created
which are overseen by people who are not team workers, and although they are a
part of a decision-making hierarchy they are not always present, therefore
decision-making is inhibited by their absence and a perceived lack of commitment
(Morgan ibid.). This impacts on a RAD-type development approach.
4.4 Communications

Recent research by Coughlan and Macredie (2002) emphasizes that communications in system development is very much a behavioural process, where human and organizational elements have an important bearing on the design. They cite a seminal study by Curtis et al. (1988) who present overwhelming evidence that strongly supports the view that effective communications is crucial in system design. They maintain that these findings still hold true.

Communication is another multi-faceted topic area. The level of communication explored here relates specifically to the effectiveness of communications during the project across both internal and external environments that influenced the project development, and the people involved. Effective communication is a major component of a successful project. It is critical to user understanding à propos, changes in working practices, accepting new business processes, and promoting co-operation within the development project. Productive communications facilitates organization wide support, it raises awareness and understanding that in turn engenders increased commitment from employees.

Effective communications require a high-level strategy and a realistic, practicable framework that provides a two-way communications channel for dissemination and feedback. Balogun and Hailey (2004) emphasize that it is important to realise that what is being communicated may not always reflect what the recipient is hearing and understanding, they say 'communication does not necessarily lead to transfer of meaning since it is the listener who creates the meaning for themselves' (2004 p167). The recipients then act on the basis of that meaning. This relates to both
written and spoken communications. They put forward four key elements for consideration i.e. the audience, the channel, the message content and the message presenter. They place emphasis on the need for communicators to be conscious of the mechanism used to convey the message, and be aware of receivers imposing unintentional interpretations upon the communiqué. Balogun and Hailey cite Lengel and Daft's (1988) representation to present some guidelines on matching audience needs to communication channels (figure 4.1).

![Figure 4.1](image)

**Figure 4.1 Audience needs aligned to communication channels**  
*Source: Lengel R H and Daft R C. cited in Balogun and Hailey (2004)*

The above figure illustrates that in complex, non-routine environments where change is present, richer forms of communications are required for those involved to develop an understanding of the situation evolving around them. Whereas in routine, more stable environments less rich forms are suitable.

### 4.4.1 Visibility and Transparency of Communication

Understanding communications is associated with the visibility of communications, what is not seen or accepted as being communicated cannot be understood. Problems with visibility can be extended to include transparency - a relation of visibility. For clarity the researcher referred to the Oxford Reference Dictionary that
defines 'Transparent' (adjective) as: ‘easily seen through; clear and unmistakable; easily understood; free from affection or disguise’ (1986, p874).

Where ineffective communications are experienced, where understanding is inadequate to convey sufficient understanding, that information becomes meaningless. Information itself is not necessary transparent, it can confuse, deceive and mislead as easily as it clarifies. Dissemination of information is often not sufficient to enable understanding by the recipients.

### 4.4.2 Shared Experiences

Cockburn (2002) maintains that there should be a reliance on informal face-to-face communications, i.e. direct links, in project environments because they engender a better level of understanding about the problems and potential solutions involved. He states that the success of communications is dependent on the sender and the receiver having a common point of reference. He further explains that this point of reference then extends to include subsequent communications that are shared between the sender and the receiver such that new concepts are constructed gradually allowing them to build new points of reference from known experiences. Those people outside the frame of reference need to be kept informed if they are to share the same level of understanding. This is similar to the view put forward by Walsham (1997), who subscribes to the belief that people’s theories about reality revolve around shared meanings. It is their interpretation of reality, their shared sense of the world. Transformation and changes that occur over time influence their perceptions by either reinforcing existing contexts and structures, or allowing the creation of new ones.

The need for face-to-face communications as a core mechanism is not argued here but where there is evidence of conflict and unrest a less expressive
communications channel, such as documentation, can be effective by removing redundant emotion. It provides the sender with an opportunity to reflect upon the message before delivery (Cockburn 2002). Although Cockburn makes many references to Extreme Programming methods the researcher posits that the above views expressed are equally applicable to the communications experienced within the case study project.

4.4.3 Tacit, Explicit and Implicit Knowledge

Where communications carrying important organizational messages are cascaded across large environments, there is a need for them to be of sufficient richness such that they incorporate communication cues which can be understood, as well as a context or domain of how they should be interpreted. It is the cues that stimulate an exchange of communications, they engender two way interactions, they allow for expression of concerns and the sharing of interpretations to create the state of 'knowing'. (Balogun and Hailey, 2004). Nonetheless, 'knowing' should not be confused with tacit knowledge. Knowing requires present activity, tacit knowledge does not, it is the activity itself that is a form of knowing. Tacit knowledge, first expressed by Michael Polanyi in 1967, represents the ability to perform tasks without necessarily being able to articulate how they are done (Beynon-Davies 2002). Polanyi states 'we can know more than we can tell' (1967, p4), he refers to a functional relation that exists between two terms of tacit knowledge, proximal and distal, where an individual can be aware of a reaction to a stimulus through experience without being able to specify the cause of the reaction itself. There is no explicit knowledge of unknown things present in tacit knowledge (Polanyi, p23). However, it is not the intention of this research study to delve into the different dimensions of tacit knowing, but rather to explore a more
current interpretation of it, that of tacit knowledge gained through understanding and experience in the context of individual knowledge that can be discovered.

Knowledge is dynamic, not all knowledge can be captured, stored and transferred, therefore it is important to focus activities/mechanisms to support knowledge action and enable effective interaction between people with different tacit powers and understanding (Walsham, 2004). Eva (2001) describes it as knowledge that is not susceptible to verbal explanation in the same way as recognised knowledge. Nonaka (2000) maintains that tacit knowledge is highly personal, it is individual know-how gained through personal experience and therefore difficult to communicate. Individuals create mental models, develop perceptions and beliefs that are taken for granted that are not easily articulated.

Whereas explicit knowledge is knowledge that is readily accessible, documented and organized (Beynon-Davies 2002), implicit knowledge is knowledge that can be gathered through communicating i.e. querying and discussion. Organizational knowledge is a crucial resource; it concerns all three types of knowledge and is managed through knowledge management. It is through the accessibility, and effective communications of knowledge that individuals acquire knowledge which can improve their performance (Beynon-Davies 2002). This is achieved through the conversion of ‘captured’ tacit knowledge into new explicit knowledge.

Carnall (2003) suggests that organizations attempt to solve problems by using tacit knowledge, but where solutions are incomplete new explicit knowledge is sought, and by doing this the tacit knowledge evolves subsuming the new explicit knowledge. Thus tacit knowledge is dynamic, it evolves through understanding and experience and needs to be captured to be of benefit to the organization.
4.4.4 Communications and Culture

Willcocks et al. put forward the theory that previous research has concentrated on looking at the issues of user 'attitudes towards the IS development being developed and the changes imposed, rather than attitudes towards and perceptions of IS personnel' (1997, p173). Their research draws the conclusion that any cultural gap is interpreted as a communications gap that can be improved through better communications. However, it is not necessarily the case that increasing the level of communications equates to providing better communications. They believe that increasing levels of communication tends to highlight communication deficiencies rather than alleviate existing problems.

In relation to requirements elicitation, the degree of understanding depends on the way the requirements are communicated, the links to detail, the richness of the data gathered, and the sharing of that data. However in complex and vast project environments, this is difficult and necessitates increased efforts of communication in order to transcend a communications gap. However, Brooks (1995) upholds the view that team communications should be diverse. He maintains that the structure of organizational communications needs to represent networks, not tree structures. Therefore communications need to be created such that they overcome the communication deficiencies inherent in hierarchical structures previously discussed.

4.5 Summary

This literature review is complementary to the initial in-depth and comprehensive study of the current status of RAD as a development approach set within the context of the IS domain. Systematic analysis driven by the issues identified
through data collection and preliminary analysis identified specific areas within the research case study that necessitated further research and deeper analysis. Consequently this chapter has presented a more in-depth and complete approach to issues of user involvement, requirements elicitation, decision-making and communication.

As previously mentioned culture is a major influencing factor in this research case study, and therefore the researcher believes it needs to be discussed in greater detail, thus it is the subject of a further chapter (see Chapter 6).
CHAPTER FIVE THE NEW IT SYSTEM CASE STUDY

5. Introduction

This case study concerned a large, complex Information System development that was being implemented within a UK Regional Government Department that, contrary to the published beliefs, adopted a RAD-type development approach. The case study is significant as it provided an interesting and atypical opportunity for the researcher to investigate and analyse the use of RAD within such a large and complex development arena and question the views and opinions about its scalability expressed in the literature.

The aim of this chapter is to describe the context of the case study scenario. It is divided into four sections such that the reader can gain an understanding of the issues involved. The first section sets out the background, historical and organizational perspectives, and explains the context of the CAP schemes, together with the rationale behind the new system and the procurement process involved. The second section presents a high-level view of the New IT System. It outlines the management and administration of the CAP schemes, their processing and their interdependencies. The third section gives an overview of the system and technical architecture that supports the New IT System. Finally, section four describes a high-level view of the development approach that was applied to the case study project. It outlines the implementation strategy and the extent to which the CAPM Business Vision was achieved.
SECTION ONE - CONTEXT OF THE NEW IT SYSTEM CASE STUDY

5.1 Background to the Case Study

This section sets out the context of the case study. It describes the background, the historical perspective and the organizational structure. The researcher's aim is to give the reader an understanding of the key issues involved. It is important to gain an understanding of the Business (CAPM\(^7\)) Vision and business objectives, which were fuelled by the rationale behind the New IT System. Additionally it is necessary to establish the significance of the RAD-type development approach and the degree to which both the RAD approach and the business objectives were achieved.

5.1.1 Scope of Case Study

This case study concentrates on the New IT System that is part of a wider Programme of Change. Therefore it is only concerned with the New IT System that deals with administration of the CAP schemes. It does not define in detail those issues outside its scope such as the other projects within the Programme of Change, or the individual minutiae of each CAP Scheme. Additionally it is not the intention of this case study to provide an in-depth analysis or criticism of the technical structure or detailed technical specification beyond the system overview description. The primary purpose of the case study is to investigate and examine the utility of the RAD-type development approach adopted, and as such fulfils one of the research objectives.

\(^7\) CAP Management- the operations side of Regional Government Department that specifically deals with CAP administration.
5.1.2. Timeline

The timetable presented in figure 5.1 below chronicles the planning and initiation stages of the project. It also shows that of the four development stages planned only stages 1 and 2 were achieved and that the 3rd and 4th planned IAD development stages of the key milestone plan were abandoned due to the internal and external pressures experienced. As can be seen the second stage continued until a new project (Single Farm Payment Project) evolved out of the New IT System project (see figure 5.14, p179). The rationale and transition into the Single Farm Payment Project is more fully explained in Section 5.4.10. p189.
## TIMELINE

### Activities

**Pre-Project Activities**
- Devolution of CAP Administration: 05/1999

**Initiation of Programme of Change Project**
- 04/1999
- IT Project Initiation Document: 09/2000

**Procurement**
- Invitation to Tender: 08/2000
- Supplier Briefing: 11/2000
- Supplier Proposals: 12/2000
- Contract Awarded: 03/2001

**Suppliers Start Project Development**

**Stage 1**
- 04/01 - 06/01
  - 'As Is' documentation of the existing system
  - 'To Be' vision of the business

**Stage 2**
- 06/01 - 04/03
  - Iterative and Incremental development
  - Cycles of requirements gathering, build and test

**Initiation of Single Farm Payment (SFP) Project**
- 04/2003

### Development Approach

**Structured Approach**
- i.e. SSADM

**Single Farm Payment Project**
- PRINCE 2
  - Documentation

### Delays/missed deadlines
- caused by internal and external factors affecting the IAD development approach
- Prolonged Stage 2 Cycle IMPACTS during stage 2 resulted in abandoning Stages 3 and 4 of scheduled development plan

---

*Figure 5.1 TIMELINE - Chronology of Case Study Project and its Pressures (This is more fully explained in Section 5.4, p171)*
5.1.3 Historical Perspective

In May 1999 under the UK Government's Devolution legislation a Regional Government Department took on the devolved functions that were previously carried out by a former Government Department. It became an independent body accountable for administering all aspects of the EU's Common Agricultural Policy (CAP) schemes across the region. As an accredited EU Paying Agency the Department has a legal obligation for administering the European Agricultural Guidance and Guarantee Fund (EAGGF) grants and subsidy payments to customers across the region through a number of CAP schemes. It is responsible for making these payments within a specified timeframe and must ensure that all EC regulations and compliance issues have been met before payment is made.

Due to the dynamic nature of the EU's agricultural policy, schemes are frequently changed with new schemes being drafted by the member states as required. Each year the EC issues a number of directives that impact on the administration and management of the CAP schemes that all Member States must comply with. Therefore CAP Management (CAPM) Division, which is the operations side of Regional Government Department that specifically deals with CAP administration, is specifically required to comply with the EC CAP schemes legislation and regulations using a framework of controls and working practice mechanisms as directed by the EC. Thus as a part of the UK as a Member of State, it is answerable directly to the EC and at risk of fines and penalties imposed by the EC that are collectively termed Disallowance. The primary objective of the CAPM Division is to make accurate and timely payments to their customer base.
In the event of a failure to make CAP payments or to meet the specified payment deadlines in accordance with EC regulations and legislation, then the Department is liable for Disallowance. This means that the EC disallows any CAP payments that go through late as a result of delayed processing, these monies are not reimbursed by the EC, Central Government has to bear the cost. Other types of disallowance include a lack of, or deficiency in key control mechanisms and a lack of key management processes. For example to deal with debt, i.e. money due to, and/or money owing from customers. These factors expose the Department to the vulnerability of significant disallowance that could result in the European Commission imposing penalties of up to £20m p.a.

5.1.4 Organizational Context

The CAPM Division of the Regional Government Department is run from a number of locations throughout the region, and consists of approximately 540 staff dealing with 100,000s of scheme payments totalling approximately £200m annually. Headquarters is centrally located and there are 3 Divisional Offices (DOs) acting as powerhouses of CAPM functions, together with 7 Area Offices (AOs) spread across the region.

Historically CAP scheme administration was organized into separate silos⁸ that dealt with specific schemes and their requirements. Each CAP scheme specifies the EC rules and conditions that detail the eligibility for its grants and subsidies. Scheme management was the responsibility of scheme specific Business Process Managers working individually, reporting to one of two Scheme Managers. It was these managers who attended to the business needs and administration of the schemes respectively.

⁸ Silo – refers to people working together in small discrete teams.
The Department has a number of links with outside agricultural organizations:

- Intervention Board – an Executive Agency responsible for receiving and accounting for the CAP Guarantee Funds. It is the UK funding agency and the UK coordinating body in respect of EAGGF.
- Ministry of Agriculture, Fisheries and Food (MAFF) – takes a lead role in the UK in both policy formation and the subsequent formulation. It provides CAPM with advice, scheme rules, administrative procedures, literature and payment timetables. Note: MAFF has since become DEFRA – Department for the Environment, Food and Rural Affairs.
- Central Ear Tag Database (CETD) – which logs all ear tags of claimed cattle throughout the UK. This information is essential for validation exercises for certain CAP schemes grants and subsidies.
- British Cattle Movement Service (BCMS) – a part of MAFF, is responsible for issuing passports to registered animals and maintaining the Cattle Tracing system (CTS) that records the identification, movement and death of cattle.
- Farming and Rural Conservation Agency (FRCA) – a government owned agency which advises CAPM on environmental programmes and grant schemes.
- Agricultural Development and Advisory Service (ADAS) – provides a range of consultancy and business services to farmers and other land users.

Data Source: Pj Ref DT 01 001

CAPM also consults with the associated farming unions and organizations\(^9\), and works in conjunction with a number of internal government departments such as

---

\(^9\) Farming Union for Region, National Farmers Union, and the Country Landowner Association.
the Departments Finance Division. Payments to customers are made by the Finance Division to allow for a separation of duties between the processing section and the paying section.

In addition to the above stated links there are a number of close relationships and dependencies that exist between the CAPM Division and internal and external parties. These stakeholders are identified and set out in figure 5.2 below:

![Key Stakeholders](image)

*Department for the Environment, Food and Rural Affairs

**Figure 5.2 Key Stakeholder of CAPM Division.**
*Source: Adapted from Pj Ref DI 10 001*

### 5.1.5 CAP Administration / Management

CAPM Division is responsible for administering 12 CAP schemes that result in payments to the farming community, and it also administers two control mechanisms:
• a Customer Details System (CD(W)) - an EU directive that member states have a common integrated system to handle all communications between CAPM and its customers.

• an Integrated Administration and Control System (IACS) - also an EC requirement which defines the common rules for the administration and control of the EC support schemes.

IACS is the principal control mechanism. It encompasses data that cross-references across a number of CAP schemes with the purpose of combating fraud. All customers must complete both an IACS form and a CD(W) form that contains all the core customer details in line with EU regulations in order to register their business if they want to receive payments of scheme grants and subsidies.

Each of the schemes, some of which are payable more than once a year, are payable within an annual cycle that coincides within a particular control period. These are aligned to a calendar window framework and some schemes operate year round enabling multiple claims to be made. In order to claim for a CAP grant or subsidy a scheme application claim form must be completed and submitted by applicants within the specified time ‘window’ of the desired schemes’ lifecycle.

The CAP schemes do not exist independently of each other, but acquiesce to a network of complex interdependent relationships. Not only are they computationally complex but are also individually dependent on specific payment windows within the annual cycle. Associated payments must conform to the EC legislation and EC regulatory control mechanisms that undergo continual change.
Figure 5.3 overleaf represents a diagrammatic view of the CAP scheme dependencies. This is a complicated diagram and it is only intended to illustrate to the reader the complex relationships that exist between the CAP schemes themselves, and the control mechanisms.

The different CAP schemes are labelled in blue running down left hand side of the figure, their processing stages are marked with ☐ symbols. To enable better understanding, one scheme has been highlighted with green symbols ☐ and all the inter-related dependencies and relationships of that particular scheme are identifiable with red lines and pink symbols ☐ to illustrate the complexities that exist between the CAP schemes.

For further clarity figure 5.4 that follows on (p154) highlights the relationships that exist between one scheme, the SCPS\textsuperscript{10} scheme, and the other schemes. This is the same scheme that is illustrated in green and red on figure 5.3 and better illustrates the complexity of the scheme relationships.

\textsuperscript{10} SCPS = Suckler Cow Premium Scheme – a CAP subsidy scheme.
Figure 5.4 Shows the Relationships between SCPS and the other CAP schemes and controls.

The above diagram is a high-level view of the most significant interdependencies and relationships of SCPS, IACS and the other CAP schemes.
In short, IACS and CD(W) information is used to validate all subsequent applications for CAP grants and subsidies within a specified year across all schemes. It necessitates an annual registration of land area and business details by all customers.

5.1.6 Rationale for the New System

As part of the Comprehensive Spending Review the Secretary of State for the region commissioned an Evaluation Report (WOAD 1998) to assess the administration and management of CAP across the section. It revealed that the performance of CAP administration was of an inadequate and poor standard such that the level of service provided was far lower than targets set under the Farmers Charter. The report further recognised that the growing complexity of the CAP schemes and control mechanisms together with the constantly changing business needs meant that the former CAPM division was not functioning effectively. It was recognised that it was unable to meet the continuing aims of the Government Department at that time to the extent that there was a serious threat of removing the CAP scheme services from the region and incorporating it within a UK provision (WOAD 1998).

Formerly the Department’s structure was characterised by a significant degree of autonomy that tended to reinforce boundaries and work against change. This was reflected by inconsistencies in work procedures and practices across the different offices that promoted a reactive focus to problem solving to the extent that planning became scheme specific rather than operating at a strategic level. Consequently, short-term fixes resulted in long-term problems that have contributed to the development of systems that address particular scheme requirements at the expense of co-ordination and systems integration.
The Evaluation Report set out a number of options and recommendations designed to inform strategic decision-making on the future of the Department at that time. From the options presented a decision was made to retain responsibility within Regional Government but to revolutionise the administration of CAP through a £16m investment programme. For the purposes of the research study this shall be referred to as the Programme of Change. The Programme was aimed at improving the effectiveness and efficiency of the administration of CAP schemes across the region, and at providing an effective one-stop service to customers using the existing network of local offices. The comprehensive Programme of Change was supported by a number of projects over a projected 3 year duration (2000-2003), these are set out below in table 5.1:

<table>
<thead>
<tr>
<th>Multi-skilled Teams Development</th>
<th>Open Plan Offices Strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Corporate People Strategy</td>
<td>BACS Take-up Strategy</td>
</tr>
<tr>
<td>Communications Strategy</td>
<td>Telecommunications Strategy</td>
</tr>
<tr>
<td>Workforce Planning</td>
<td>Developing Leaders</td>
</tr>
<tr>
<td>Geographical Information System</td>
<td>Customer Focus</td>
</tr>
<tr>
<td>Performance Management</td>
<td>New IT System (Research Case Study)</td>
</tr>
</tbody>
</table>

Table 5.1 Programme of Change Projects

Investigations into developing a replacement IT system started early in 2000. An Initiation Stage was undertaken to establish whether the project was feasible and a Full Business Case was conducted to define the problem and assess the most appropriate option. This outlined an analysis of the Strategic Case, the Economic Case, the Financial Case, the Commercial Case and the Project Management Case against the investment objectives, strategies, benefits and risks involved to produce a shortlist of options for appraisal. The preferred option of using an external supplier with in-house support and maintenance was selected then put forward and approved by the Project Board.
The UK Government Treasury agreed to fund 75% of Programme of Change costs from their 'Invest to Save' budget which is aimed at encouraging cross boundary working to ensure more efficient and effective public services. Hence a Project Management Team was recruited specifically to manage the project whose Project Manager reports to the Programme of Change Manager, who in turn reports to the head of the CAPM Division. This is a formal reporting procedure using a Project Board and Programme Board to facilitate the governance of the project as set out in figure 5.5 below:

Figure 5.5  *The New IT System Project Reporting Structure*

The above figure shows the formal reporting structure between the Programme of Change project and the IT project. It is not representative of the actual project teams and their composition. The New IT System became the IT enabling part of
this Programme of Change and is the focus of this case study research and hereafter is referred to as the New IT System.

5.1.7 Business Objectives

- To provide the Department with an IT system for the management of the CAP grants and subsidies for the agricultural community across the region
- To provide an IT system that enables the Department to change business practices and business rules.

The key principles of the New IT System reflect an integrated customer centric solution through redesigned and standardised processes that involve automated data capture, data validation and speedier payments to customers. Additionally to incorporate flexibility, and an improved accessibility of information to both staff and management through a Document Management System (DMS) to provide a consistent service across the region in line with the CAPM Business Vision.

In line with the CAPM's developing vision a Document Management System was created to store the project documentation. The DMS was designed to ensure that correct project information and documents were available to management and other users via a comprehensive enterprise wide system. It offers the required flexibility and improved accessibility of information to staff and management as well as providing a consistent service across the region.
5.1.8 Procurement Process

The Project Board decided that due to lack of skills in-house and insufficient expertise the development of a New IT System should be undertaken by an external supplier who would work in partnership with CAPM to successfully meet the proposed system objectives, but that support and maintenance of the CAP schemes would remain in-house, consequently it was put out to tender.

In accordance with EU Legislation an Invitation to Tender was placed in the Official Journal of European Communities resulting in 76 expressions of interest from external suppliers who were all provided with background details of the Regional Government Department and the New IT System and asked to complete a questionnaire. A short list of 6 was drawn up from the 35 suppliers who returned the completed questionnaire, and these were then issued with a high-level Requirements Specification for the replacement system, and invited to submit a proposal addressing these defined requirements.

From a final shortlist of 3, a supplier was put before the Project Board and Permanent Secretary for approval and the suppliers entered into a contractual agreement with the Regional Government Department to design, develop and implement the New IT System that would be in keeping with the aims of the Programme of Change. The supplier was selected on the basis of their experience in the field of systems transformation and of developing customised software and hardware, and approved by the Project Board and Permanent Secretary. The project arena is spread across four floors of a centrally located government building where both the Department and the Suppliers are co-located together for the project duration. A project structure was created that involved small integrated
teams comprising of both Departmental staff and Supplier Personnel that were specialist and subject specific according to need. (See table 5.1, p156 and figure 5.5, p157)

5.1.9 Project Community

The New IT System project community reflects those people within the immediate New IT System project and those that are involved also with the Programme of Change project. This is defined in the table 5.2 below and pictorially represented in figure 5.6, reflecting both environments. It sets out clearly the Senior Management Team and the Core Project Team that are referred to throughout the case study:

<table>
<thead>
<tr>
<th>Project Community</th>
<th>Project</th>
<th>Project Team</th>
</tr>
</thead>
<tbody>
<tr>
<td>Head of CAPM Division</td>
<td>Programme of Change</td>
<td>Senior Management Team</td>
</tr>
<tr>
<td>IS Manager</td>
<td>Programme of Change / New IT System Project</td>
<td>Senior Management Team / Core Project Team</td>
</tr>
<tr>
<td>Client Project Manager</td>
<td>Programme of Change / New IT System Project</td>
<td>Senior Management Team / Core Project Team</td>
</tr>
<tr>
<td>CAPM Business Managers (Includes CAPM Business People)</td>
<td>New IT System Project</td>
<td>Core Project Team</td>
</tr>
<tr>
<td>Supplier’s Project Manager</td>
<td>New IT System Project</td>
<td>Core Project Team</td>
</tr>
<tr>
<td>Supplier’s Personnel</td>
<td>New IT System Project</td>
<td>Core Project Team</td>
</tr>
<tr>
<td>i.e. analysts, developers etc.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Table 5.2 Project Community*

Figure 5.6 defines the relationship of the New IT System to the Programme of Change and clearly identifies the Senior Management Team and the Core Project Team.
The above figure is designed to illustrate the relationship of the New IT System to the Programme of Change, and also to describe the Senior Management Team and the Core Project Team that are referred to in the case study. It is not a true representation of the project’s structure, where the CAPM business managers would be placed alongside the Supplier’s Project Manager.
SECTION TWO - THE NEW IT SYSTEM OVERVIEW

5.2 Introduction

This section provides a high-level overview of the New IT System and the Generic Process. It sets out the CAPM business vision and objectives. It defines the new processing activities and introduces business rules and workflow concepts that are further discussed in Section 5.2.3.

5.2.1 High-level Overview of the New IT System

It is necessary to provide a holistic overview of the New IT System such that the issues impacting on the development approach adopted can be fully understood by the reader.

The New IT System provides a number of integrated IT services to support the CAPM business model i.e. the proposed Generic Process. The CAPM Management Division deals with the CAPM business. This is an application processing function where customers submit an application form for a grant/subsidy under a number of common EC agricultural policy schemes. The CAPM business evaluates the applications against complex rules and entitlements, communicating with customers where appropriate and calculating the eligibility and size of any payment due.
The desire to move away from the former CAP systems emphasis on silo structure, together with the identification of significant similarities between the scheme processes and payment mechanisms promoted the development of the Generic Process Model. It amalgamated areas of the separate schemes through a common non-specific method of processing and, provided a centralised view of all business processes against all schemes. The figure 5.7 below illustrates the common processes across the schemes:

![Figure 5.7 Generic Process Model](source: Adapted from Project Ref DT.01.001.01)

5.2.2 CAPM Business Vision

The primary CAPM aim was to improve CAP administration through projecting a CAPM Business Vision to be achieved through the following objectives:

- a higher quality and delivery of service to customers
- a reduced cost of CAP administration
- to meet EC rules and requirements
- to avoid disallowance and penalties, incurred through late payments to customers, that are payable by the Department.

The generic model would not only process the present CAPM schemes more effectively and efficiently but facilitate the introduction of new schemes and changing business needs without significant redevelopment, such that CAPM personnel are able to respond to EC changes and the introduction of new schemes.
5.2.3 CAP Form Processing in New IT System

The customer submits signed application claims forms, that were pre-printed with their individual details and sent to them, detailing any amendments. In synopsis, these are then scanned for data capture and data extraction and receipted\(^\text{11}\) into the New IT System. The original form is sequentially archived and the image and ‘data set’ of that application form is stored in a database and passed through to the workflow process that starts the CAP scheme processing. The workflow is the computerised automation of the business processes standardised across all schemes that enables the scheduling, routing and delivering of information automatically. The nature of workflow systems involves the processing of one application through a number of steps necessitating sequential business processing i.e. completing one step at a time and then releasing it back into the workflow.

A Business Rules Engine that contains all the CAP Schemes and EU regulations and requirements, identifies and activates generic and scheme specific business logic rules that represent the business processes relevant to that particular application claim. These are launched dynamically during run time, enabling automated workflow processes to apply acceptance and validation and to calculate payment procedures against each customer’s application claim form(s).

‘Perfect claims’ without errors or anomalies, and which satisfy the relevant business rules require no manual intervention. They pass through for payment processing via an interface to the current finance systems Accounting Matrix. The

\(^{11}\) This is an official requirement which has both legal and payment implications and is the Record Receipt step of the Generic Process.
following figure 5.8 below is representative of a high-level overview of New IT System:

![High-level Overview of New IT System](image)

**Figure 5.8  High-level Overview of New IT System**  
*Source:* Adapted from Pj Ref: ST1.2.1.2.2

However, application claim form data sets that fail to satisfy the business rules at any point in the Generic Process are not allowed to pass the Validation Process, they are placed into a work queue within the workflow procedure and wait for the attention of the multi-skilled teams, who manually resolve the queries raised by the failed business rules and return the application claim back into the workflow process. This continues until the application is successfully accepted or rejected as not eligible for payment.

Finally, because a number of the schemes have interdependencies, system checks are made to verify if there are any dependent applications waiting in Data Validation for this application to complete the payments steps. Figure 5.9 below
represents a high-level overview of the New IT System from the completion of an application form by a client to the payment process.

If an application is dependent on data from another source or if the scheme payment window is not yet open it is sent to a holding area until that data is available. Similarly those processes that can be performed at any time can be allocated to a specific work queue. Figure 5.9 below illustrates the workflow of the main processes.

**Figure 5.9 Workflow of Common Processes**

*Source: DJ.03.01*
5.3 Introduction

This section describes the adaptive architecture design adopted for the New IT System that supports the new CAPM business model. The New IT System architecture is separated into Logical Elements, Physical Elements and Technical Software Support Elements and described by the Lead Architect as set out below.

5.3.1 Logical Elements

The logical cycle defined the logical technology components and IT mechanisms required to deliver the conceptual architecture. These are set out below and are visually illustrated in figure 5.10.

- The System is based around a 3-4 tier architecture with most client machines receiving software through a browser (e.g. screens).
- Front end HTTP servers, distributed at the main site provide local copies of large software elements to maximise network efficiency.
- A mid-tier application layer contains business logic and rules, workflow, document management and GIS\(^\text{12}\) functionality.
- A back end layer (Database Tier) contains the business data. Interfacing and batch operations are also handled at this tier.

\(^{12}\) The GIS referred to is not a part of this research study, it is one of the other projects within the Programme of Change.
5.3.2 Physical Elements

The physical cycle translated the preferred logical architecture into an implementable physical architecture and specified the IT techniques, standards and products to be used as detailed below:

- The client machines are OSIRIS standard build machines with MS Internet Explorer 5, through which the client software is delivered to the user. Elements do not necessarily run in the browser window but are downloaded through it.
- The front-end layer, based around MIS IIS are based on a number of Compaq DL360 servers distributed around the organization.
- The mid-tier and database tiers are based on a number of servers using a storage area network (SAN) to provide fast, resilient storage of approximately 1TB logical capacity. The database servers are clusters.
The diagram of the overall system, maps business process elements to architecture elements and represents the mapping of the generic process to the technology. This is illustrated below in figure 5.11.

![Diagram of the Physical Architecture](image)

**Figure 5.11 Illustration of the Physical Architecture**  
*Source: Pj Ref ST12.2.1.3*

The system is a multi-tier distributed system based on component technology, the FileNET DMS and workflow engines. Client functionality is delivered through Microsoft's Internet Explorer web browser with the exception of the scanner which requires locally installed software. It is built upon the existing OSIRIS/ISIS standard desktop that includes MS Office suite and MS Outlook. The middle tier includes document management services, workflow services and transaction services that provide scalability and high availability. Middle tier servers are located at each of CAPM main sites replicating data between them for resilience. The database tier uses Microsoft SQL Server Enterprise Edition supported by scheduling and reporting software and the server operating system is Microsoft Windows 2000.
5.3.3 Software Support Elements

There are a number of commercial software components that are key to the system; these are set out and explained in Table 5.3 below:

<table>
<thead>
<tr>
<th>Software</th>
<th>Role and Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>MS Windows 2000 Advanced Server incl Cluster Services</td>
<td>Operating system security administration and HTTP server</td>
</tr>
<tr>
<td>SQL Server 2000 EE</td>
<td>Relational database providing line of business data, GIS data, document management system data and workflow data repository.</td>
</tr>
<tr>
<td>FileNET Panagon eProcess</td>
<td>Business process automation (workflow)</td>
</tr>
<tr>
<td>FileNET Panagon Content Management</td>
<td>To aid the storage and retrieval of documents. Includes features that provide an audit trail and security.</td>
</tr>
<tr>
<td>ILOG JRules Business Rules Engine</td>
<td>Allows business logic to be defined directly by the business in an English-language like format. Business rules can be dynamically loaded at runtime to provide scheme specific logic within a standard business process.</td>
</tr>
<tr>
<td>Kofax Ascent Capture</td>
<td>A suite of products that provides image capture and processing. Bespoke elements are used to release documents into workflow and into the document management system.</td>
</tr>
<tr>
<td>Neurascript</td>
<td>Provides enhanced character recognition within the overall scanning process.</td>
</tr>
<tr>
<td>ESRI GIS</td>
<td>The presentation of and calculations involving geographic data.</td>
</tr>
<tr>
<td>Seagate Crystal Reports</td>
<td>Production of letters and reports (generally in electronic format)</td>
</tr>
</tbody>
</table>

Table 5.3 Key Software Components of New IT System
Source: Pj Ref: ST1.2.1.2.2

The HTTP servers are based at Divisional Offices and run Microsoft Internet Information Server and FileNet services.
SECTION FOUR - RAD-TYPE DEVELOPMENT APPROACH

5.4 Introduction

This section compares the development strategy that was planned with the actual strategy that was applied to the New IT System project. It identifies the major external and internal impacts that impacted upon the RAD-type development approach.

It further describes the System Development Lifecycle that sets out the build, test and implementation practices applied, as well as the processes and procedures used for the design and development of the new system together with the implementation strategy. Additionally, it explains the transition of the New IT System project into the Single Farm Payment project.

It is the aim of the researcher to emphasize aspects of the development of the project in relation to the utility of an IAD/RAD approach, not to focus on the product itself - the New IT System. All of the issues raised are examined and discussed further in the analysis Chapters 7 and 8.

5.4.1 SSADM to IAD

Text books refer to SSADM (Structured Systems Analysis and Design Methodology) as the prescribed Government standard for Information Systems development. They state that it has been mandatory since 1983 (Weaver et al. 1998) but this does not appear to be true of the New IT System Project, it was not prescribed for the Programme of Change, and the Department were not guided to
use it. Evidence suggests that prior to procurement this structured approach was applied by the Department to obtain the initial high-level Requirements Specification. For example process models were used to document the ‘As Is’ scenario of the existing system and to create the initial high-level ‘To Be’ vision of the business requirements from which a Requirements Catalogue was formulated. This identified the functional and non-functional needs of the proposed new system. These were quality, usability, reliability, performance and supportability, in other words the integrity of the system. Data Flow Diagrams (DFDs) were created for each CAP scheme that emphasized a commonality across the administrative processes of the schemes facilitating the conceptualization of a Generic Process Model. It enabled the integration of the common activities across the CAP schemes that provided a basis for the project’s development activities.

Once the procurement process had been completed the chosen Suppliers adopted their preferred method of Iterative Application Development (IAD). Traditionally they used this type of approach for projects, where the end point was not very well understood and where there was no fully defined specification as was the case with the New IT System project. The volatile nature of the CAP schemes due to the continual amendment of the EC regulations and their legislation, together with the introduction of new policies means that it was not possible to define fully the changing business needs of the Department.

5.4.2 IS development Project – Size and Complexity Issues

The case study project is described as both large and complex. Large in terms of the financial investment, an initial estimate of £10m allocated from the £16m budget provided for the Programme of Change. Its size is also reflected by the proposed 2-3 years duration of the project. Most small to medium projects are
described in terms of weeks or months. Other measurements include the scope of the project personnel involved – a core team of 50+, and the extent of the Department's customer base across the region. Complexity in IT projects is traditionally measured in terms of lines of code. There is a difficulty in calculating accurately how complex a project may be, especially if the system involves new or re-engineered business processes where it is not possible to identify all business requirements upfront. However in the context of this case study complexity refers primarily to the interdependent relationships that exist between the CAP schemes and the EC legislation, and their dynamic nature. A further indication of the complexity of the new system was the volume of business rules aligned to business processes that were developed – in excess of 4,000, that dealt with the highly complex computational calculations necessary for the scheme payments.

The issues raised in the literature surrounding the scalability of a RAD type approach reflect the difficulty of managing the development of large projects. However, the case study project has demonstrated how larger development projects can successfully be divided up into manageable development modules. The RAD type development approach involved small integrated project teams pulled from the core team of 50+ of the developers and the organizational people co-located on the same site, working closely together throughout the project. These project teams were specialist / subject specific according to design and development needs. Thus as the project evolved so did the teams.

5.4.3 RAD type Development Approach – Iterative Application Development

The Suppliers applied an Integrated Architectural Framework (IAF) as the New IT System Project architecture method. The IAF is described by the Suppliers as an industry-leading approach to architecture that forms the basis of a global IEEE
standard. It is the IAF that links the IAD method of the system architecture to the software architecture supporting the high-level business principles. They put forward previous successful development applications across the globe as evidence of its suitability. These included facilitating a modular approach involving legacy systems for a market-leading electrical company in Sweden, a development project for a Central Securities Depository in Norway and replacing business critical system for one of the largest communications companies in the UK.

An IAF approach is driven from the business strategy and has the advantage of not being bound to specific technologies. It took the current business processes and IT provision and transformed them into a more specific 'To Be' solution. However although it provided the context in which the New IT System would operate it was the application of Use Case modelling and JAD workshops that were used to drill down into the detail of the requirements. This provided the functionality of the business processes as identified through the initial high-level Requirements Catalogue. Figure 5.12 below summarises the IAF approach.

![Figure 5.12 Overview of the IAF Approach](source:DQ.03.001.01)
Additionally because the New IT System environment was acknowledged as being an interactive project, the Iterative Application Development (IAD) approach was justified by the Suppliers as the most appropriate project delivery lifecycle. They reasoned that it would promote a controlled and structured but flexible development environment aimed at providing incremental development and phased delivery.

They believed this approach offered all the main benefits of a Rapid Application Development (RAD) type approach that was suited to the uncertainty of, and continually changing business requirements of the development environment. It concerns iterative development and incremental delivery but without the problems of creeping feature scope and overrun that are associated with a RAD approach.

The IAD development involved the same main features of RAD, i.e. JAD workshops, timeboxing, prototyping and intensive user involvement throughout the project. The overall strategy was described in terms of iterative development cycles and incremental delivery, which they maintain is an increasingly used approach for system functionality. Furthermore it corresponded to the 'As Is' to 'To Be' concept favoured by the Department.

5.4.4 Case Study Project and the Application of a RAD approach

As it was The Supplier’s own IAD development approach (a vendor specific method) that was adopted, this section sets out the degree to which this RAD type development approach has, in fact, been applied to the research case study project. As mentioned previously the DSDM Consortium have established 9 fundamental principles (1994) that they consider to constitute a RAD approach - these are set out in table 5.4 below.

175
1 Active user involvement 6 Changes are reversible
2 Teams enabled to make decisions 7 High-level requirements (base-lined)
3 Frequent delivery of products 8 Integrated testing during lifecycle
4 Fitness for business purpose 9 Stakeholders collaboration/co-operation
5 Iterative & incremental delivery

Table 5.4 DSDM 9 Principles

In table 5.5 below these 9 principles have been aligned against the case study project to verify that a RAD type approach was being applied across the development project.

<table>
<thead>
<tr>
<th>DSDM PRINCIPLES</th>
<th>Case Study Project</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Year 1</td>
</tr>
<tr>
<td>1 Active user involvement</td>
<td>Yes/No</td>
</tr>
<tr>
<td>2 Teams empowered to make decisions</td>
<td>No</td>
</tr>
<tr>
<td>3 Frequent delivery of products</td>
<td>Yes</td>
</tr>
<tr>
<td>4 Fitness for business purpose</td>
<td>No</td>
</tr>
<tr>
<td>5 Iterative and incremental delivery</td>
<td>No</td>
</tr>
<tr>
<td>6 Changes are reversible</td>
<td>Yes</td>
</tr>
<tr>
<td>7 High-level requirements – base lined</td>
<td>Yes</td>
</tr>
<tr>
<td>8 Integrated testing during lifecycle</td>
<td>No</td>
</tr>
<tr>
<td>9 Stakeholders co-operation/collaboration</td>
<td>No</td>
</tr>
</tbody>
</table>

Table 5.5 DSDM 9 Principles as Applied to the Case Study Project

There is significant evidence that the first 8 of the principles that constitute a RAD process occurred during the Project, however although co-operation was achieved with the majority of stakeholders, a serious constraint on the RAD development process was the inflexibility of the EC as a major stakeholder. Conformity to EC legislation and regulations where the ‘fit for purpose’ requirement reflected almost 100% of business needs impacted on the timeboxing concept and the de-scoping elements of a RAD development process that proved difficult to achieve. Business managers were reluctant to sign off development work if it did not enable them to satisfy the business needs of their schemes and this led to development delays.
As put forward in the literature review a successful RAD approach necessitates cultural and managerial changes. In this particular case study a radical shift in the mindsets of ‘organizational’ people was needed to promote the Generic Process model. Evidence exposes the difficulty that these people had in moving away from their previous ‘silod’ attitude to buying into the Generic Process concept. This was highlighted by difficulties experienced with decision-making.

5.4.5 Project Development – Major Impacts

It is interesting at this stage in this case study presentation to look at the initial Milestone Plan, figure 5.13, created by the developers in the initial stages of the project and contrast it with a Key Milestone Plan that the researcher has created which highlights all the major events that impacted upon the IAD development process.
As can be seen the initial plan was composed of four key development stages, each of which had a number of phases involving timeboxed iterations. Essentially the developers envisaged completing the initial development work for the CAP schemes allocated to the New IT System during stage 2, and then revising and modifying the system and incorporating new business needs during iterations in stages 3 and 4. This development approach conforms to the RAD characteristic of iterative and incremental delivery and phased development. Therefore it can be surmised that during the initial planning stages, a RAD-type development approach was indeed planned and anticipated.
However the planned iterations did not occur. An amended Milestone Plan that illustrates the factors that impacted upon the project is shown in figure 5.14 below:

![Figure 5.14 Amended Milestone Map showing External and Internal impacts on the development project. (not to scale)](image)

Stage 1 was completed as planned but stage 2 was severely impacted by a number of external impacts and internal constraints that resulted in development delays, missed deadlines and overrun of the project into a third year. Due to these problems payments due for 2002 were significantly delayed, and the New IT System was not operational in time to meet the October 2002 payment deadlines. As the figure illustrates, project development was adversely affected by the outbreak of a Foot and Mouth crisis. More specifically, the Foot and Mouth crisis occurred during the initial stages of the project resulting in the reallocation of key
staff to assist with the crisis, this caused an initial project delay that is estimated to have been about 6 months. Consequently this had a knock on effect that contributed to problems with testing and software release schedules that were ultimately pushed back and fuelled the creation of a joint testing environment. These issues are analysed in-depth in Sections 8.3.2, p299 and 8.3.3, p301.

In addition the New IT System had to take on board and accommodate complex scheme changes via EC regulatory directives that impacted during this period necessitating additional development. This subsequent development added a further burden on the resources that were already stretched by the Foot and Mouth crisis. An added complication was the unforeseen parallel introduction within the Regional Government Department of a new finance system for automated payments to customers, which also required extra development resources that served to further delay the projects development.

Furthermore there were three occasions where changes that occurred during the project impacted noticeably upon development arena (see figure 5.15). Firstly, a move to relocate the project environment due to considerable pressure and demand for workspace at the original site was a considerable, albeit short term, disturbance. Although measured in weeks it caused a major physical disruption to the project. Additional work involved the removal of, and re-instalment of kit, servers and equipment together with the necessary network upgrades to the Divisional Offices. Even though it had benefits in terms of accessibility for the project personnel, it also had a big effect on the development resources available to do development work.
Secondly, an initial change in project personnel is placed approximately at the end of year one when a number of the outsourced Suppliers were replaced. It primarily involved those concerned with the high-level analysis and design activities when the project moved into the next stage of development. At this time a more flexible approach was adopted towards project scope of the high-level system requirements in terms of design and development parameters.

The third change also concerned a change in project personnel that occurred at the end of the 2\textsuperscript{nd} year when the project experienced overrun extending the development into a 3\textsuperscript{rd} year. In spite of this overrun the project moved from Development and Delivery Mode into the Support Phase as had been initially scheduled, even though development on the new IT system was still on going. The in-coming personnel shifted the focus away from a technological one, to one of coordinating and controlling in terms of management that ran parallel to the ongoing development. The project moved into a more concentrated holistic management phase. Associated with this was a restructuring within the CAPM.
Department where people were reshuffled within the project environment to accommodate this shift in project direction. At the end of year 3, a new project, Single Farm Payment Project, took over to accommodate radical legislative changes introduced by the EC – this is still on-going.

5.4.6 System Development Lifecycle (SDLC) Adopted

The IAD development is aligned to a RAD approach as it compresses the analysis, design, build and test phases that are associated with the traditional ‘Waterfall Model’ lifecycle into a sequence of short, iterative development cycles. Consequently the project development was broken into development modules that involved process definitions which represented the analysis stage, a module specification that represented the design stage and a series of timeboxed mini iterations that involved a number of software build, test, release cycles. See figure 5.16 below:

![Diagram of Iterative Development Cycle](image)

Source: adapted from Project ST2.2.1.50, 2001

*Figure 5.16 New IT Project Development Cycle
Source: Adapted from Pj Ref DQ 01.001.01 * User Acceptance Testing
As the above figure illustrates there were four components to the SDLC namely design, development, testing, and delivery. The SDLC is iterative in nature, during the development period several deliveries of software occurred in the form of software releases. The releases corresponded to the predefined business process functionality; these were passed initially into the integration test environment and then promoted into the system and UAT environments. The final software release was made into the live production environment. Wherever failure occurred throughout this development lifecycle, test incident reports (TIRs) were raised and issues were fed back into the IAD process during subsequent development iterations.

Although this iterative approach to software development was designed to provide early validation by the users and the business analysts through testing of the modules being developed, this was not always the case. The severity of the Foot and Mouth crisis, and issues of the availability and allocation of the right users meant that it was not always the right users who were involved\(^\text{13}\). The intention was to provide flexibility to incorporate user feedback and deal with any new or changing EC requirements within the volatile business environment – a key goal of a RAD type approach that was also in line with the CAPM vision. Therefore it was envisaged that from early on in the development process parts of the final system would be visible to both CAPM and all the development teams. This provided the ability to handle new or changing requirements in a more effective manner than the more traditional Waterfall development lifecycle would have allowed, and was aimed at improving the end product by reducing the risk factor. However this did not occur during the early stages of development, and it was not until the development process matured that this in fact did begin to happen.

\(^{13}\) The issue of right users is fully explored in Chapter 4, section 4.1, p125.
As stated previously, the original high-level requirements for the New IT System were described in the Invitation to Tender that was released to prospective suppliers during the procurement process. These were then further analysed through a series of high-level meetings between the Suppliers and the Department that established a list of development activities and subsequently a Project Development Plan. This was supplemented through JAD workshops activities that were employed to identify the high-level functions, data and interface requirements for each individual CAP scheme, thus development became scheme specific.

The Supplier's business analysts created detailed use cases of the new system. Each use case described the actors and action flows within each process. They were positioned on the ‘To Be’ process chart together with other identified associated use cases, their trigger events and non-functional requirements. Activity diagrams were used to describe the sequencing of activities within the use case that illustrated the flow of the activities between different technology items or packages being used. A Business Rules Engine was used to manage scheme specific business rules aligned to the Generic Processes that accommodated the complexities of the schemes. This is more fully covered in Section 5.4.7.

In synopsis, an end-to-end description of a CAP scheme development process began with the identification of scheme specific requirements through a series of JAD workshops. A scheme development schedule was separated into the stages of the Generic Process i.e. Receive & Lodge Claim, Validate Claim, Calculate Payment, Authorise Claim, Notify Claim and Make Payment. Schemes were then developed separately through these stages of the Generic Process.
5.4.7 Business Rules Logic

Business Rules logic was used to represent the CAPM expertise and knowledge of the business processes together with the legislative requirements of the EU Regulations. Pieces of business logic were precisely defined, described in concise statements and captured within Business Rules that were then catalogued and organized into Business Rules Sets relating to the CAP schemes. However, where rules were common across the CAP schemes they were contained within a Generic Rule Set rather than be reproduced for every single scheme.

Business managers identified and defined the business rules of a scheme in a high-level document that was subsequently presented to the Business Rules Team who re-wrote them in high-level Pseudo English rules that were subsequently then converted to lower level Business Action Language (BAL) rules and entered into a business rules engine. A Business Object Model (BOM) then maps the underpinning Java code, written by Supplier’s developers, that allows actual Java code to be executed when business rules are evaluated during run time. It is a top down design and a bottom up development exercise and is illustrated below in figure 5.17.
ILOG is a software development component that consists of a common set of tools, it incorporates an integrated development environment for developing and debugging business rule applications known as the Rule Builder. The Suppliers use an ILOG editor, a piece of software, to create the rules in the ILOG intermediate language; these rules are then stored for change control and reuse purposes. On completion of unit testing, the business rule is exported from ILOG in its compiled form, and stored in a database table on the business tier server and loaded dynamically as appropriate during run time.

5.4.8 The Business Rules Concept and Development Process

As defined in the above section, business rules represent pieces of business logic that are precisely defined, described in concise statements and captured within Business Rules and stored in a database. They are organized into Business Rules Sets per scheme representing the stages of the Generic Process. Each business rule contains an action that gets performed when a condition is met. In other words business rules are concerned exclusively with a business action that needs to be performed and the circumstances that trigger that action. In this way a
business rule represents a part, or parts of a process of a particular scheme that satisfies a specific business need.

Simply put, a business rule may be thought of as a statement using an IF, THEN structure. IF represents a condition and THEN details an action. When the condition is satisfied, the action is executed.

For example a business rule to check whether an application claim form is an IACS application:

\[
\textbf{IF} \quad \text{This is an IACS application} \\
\textbf{THEN} \quad \text{Set IACS application to true}
\]

The outcome: the application claim form will be marked as an IACS application.

This is simplistic example of a business rule aimed to demonstrate the process. The business rules process is the subject of a conference presentation at Business Rules Forum 2005 (Berger², 2005).

5.4.9 Implementation Overview

At the project planning stage there was an initial proposal by the Suppliers to rollout the Generic Process through a ‘Big Bang’ implementation regardless of their individual payment deadlines. The original thinking behind this idea was the concept that the Generic Process would accommodate all the schemes. The Suppliers reasoned that if they were all generic then the number of schemes and related deadlines was immaterial. However this early view did not take into account the logistics of either the business time scales, not all schemes were needed at the same time, nor the difficulties of resourcing such an approach. The business

\[\text{IACS = Integrated Administration and Control System (IACS), an EC requirement which defines the common rules for the administration and control of the EC support schemes.}\]
knowledge that was needed to transfer schemes onto the New IT System was in the hands of a very small number of people who were also attending to Business as Usual activities. Additionally, at this early stage of the project there was a lack of understanding, and underestimation of the development tasks involved. The Suppliers were not fully aware of the complexities involved with the interdependent relationships that exist between the schemes.

The Implementation Strategy that was subsequently produced was a phased scheme implementation approach. Consequently there was a phased implementation of the schemes from the Legacy System to the new CAPIT system’s platform and technology. IACS was chosen as the most logical CAP scheme to be processed first because it contains the base and controlling data that is required and accessed by the majority of the other schemes. The rationale behind this choice was the necessity to develop a series of interfaces across the other schemes to accommodate the IACS inter-dependencies that existed. IACS is also the largest scheme and therefore should have allowed faster integration of the remaining schemes following it. However, this approach had unforeseen consequences. When IACS development work became problematic and suffered delays and setbacks, it had a domino effect that cascaded across the other scheme developments with related interdependencies.

Implementation was also affected by EC Directives. A recent EC Directive radically altered the administration of the CAP schemes (see section 5.4.10). It required fundamental changes in the way the CAP schemes operated. This change impacted across the development and implementation of the schemes being prepared for the New IT System. The changes involved meant that a number of the schemes would cease to exist as individual schemes from 2005. Consequently
a critical assessment was made as to the viability of continuing existing development and of new scheduled development onto the New IT System in terms of investing time and effort. A high-level decision was made to maintain the schemes using the Generic Process, but to halt new development on schemes waiting to go onto the Generic Process that would cease to exist. These were transferred into a new project that evolved out of the case study project. The Single Farm Payment Project was aimed at accommodating the new objectives of the EC changes. Thus at the end of the third year the majority of development activities were concentrated on the new project objectives.

5.4.10 Transition from New IT System to Single Farm Payment Project

In June 2003 EU Farm Ministers adopted a reform of the Common Agriculture Policy that was timetabled to enter into force March 2005. The reforms represented a fundamental change in the way CAP grant and subsidy payments are awarded. Grants and subsidies would be decoupled from production based payments via the existing CAP schemes towards a Single Farm Payment provision that would replace the current schemes administration. Consequently development on the New IT Project was restricted to the current development objectives at that time. A new project – ‘Single Farm Payment’, was initiated to accommodate the CAP Policy changes and accommodate those schemes affected that were no longer destined for the New IT System. This research study concentrates on the case study as set out it does not take account of the new Single Farm Payment project.
5.4.11 Project Realization

The purpose of the New IT System was to meet the project aims and fulfil the requirements of the CAPM Vision objectives. The resultant New IT System is described as a hybrid system. The Generic Process represents an integrated solution through redesigned and standardised processes that involve automated data capture, data validation and speedier payments to customers. The links between the New IT System, the Legacy System and the New Finance System have enabled the Department to meet the fixed EC payment windows. A further measure of achievement is seen as the ability to make scheme payments on Day 1 of schemes opening. This is the first time the Department has achieved this in the history of the schemes. The ability to meet the payment deadlines provides a higher quality and delivery of service to customers and conforms to the requirement to avoid disallowance and penalties, incurred through late payments to customers.

The utilization of business rules provides the Department with the ability to accommodate changes in business practices and scheme rules that promotes a more efficient use of staff and offers a better service to customers. Thus appropriate employees are able to meet changing and new business needs and conform to the EC legislation and EC regulatory control mechanisms. However, this only applies to simple straightforward regulatory changes such as modifying payment dates or percentage values set within a business rule. For more complex interwoven changes whose impact cascades across a number of inter-related schemes, the need still exists for specialist IT knowledge that the Departments' people do not have. Consequently there is still a heavy reliance on the developers that is seen as a potential weakness.
The New IT System project was mainly funded by the Invest to Save Initiative and consequently an objective relates to benefits realization in terms of savings both tangible and intangible i.e. cost and effort. However, it is difficult to stipulate the less tangible savings as it is not possible to directly relate specific areas of cause and effect because the level of granularity is not visible to demonstrate those savings. Nevertheless it is recognised that the organizational effort required to process the schemes is considerably reduced and additionally the CAPM organization has reduced its operating costs significantly over the last three years. Project documentation reports savings of £1.6m of benefit since implementation that represents the 'value for money' criterion against which the project's success was measured. However if measured against more traditional measures of success i.e. 'on time' and 'in budget' it would be considered a failure.

Finally, it must be stated that although this particular UK region did not succeed in meeting all the EC targets during the development period, none of the other UK regions were successful either, they all experienced payment difficulties. This suggests that the development of the new system may be seen as a contributory factor but was not solely to blame. The scheme amendments imposed by the EC that caused significant changes in the key business needs had a critical impact across all the UK regions.

5.4.12 Summary

This chapter described the context of the research case study; it set out the background, historical and organizational perspectives and explained the context of the CAP schemes. It presented a high-level view of the New IT System, outlined the management and administration procedures, and further defined the system's technology and architecture. It gave an overview of the development approach
and implementation strategy that were applied to the case study project. Finally it discussed the extent to which the project aims and vision have been achieved.
6 Introduction

The purpose of this chapter is to define and characterize both the main research themes, and the development factors that impacted on the research case study project that have been aligned to the 9 DSDM principles. The main themes have previously been distinguished in terms of people management, cultural constraints and human factors (Martin 1991, McConnell 1996, Sadler 1996, Hirschberg 1998, Jones and King 1998, Tudhope et al. 2001). These factors represent different forms of culture, and culture is an element that literature regards as fundamental to the application of a RAD-type development approach. The development factors referred to denote those development factors identified as impacting significantly upon the research case study project, for example user involvement, requirements elicitation, user expectation, communications, decision-making and testing. These areas were also seen to have been influenced by cultural issues.

Culture is a complex topic area; here the question of culture is considered in terms of individual, group and organizational contexts that are relevant to this research case study. However organizational culture is axiomatically related to organizational structure, which for this IS project arena concerns a public sector environment that is bureaucratic in nature.

Essentially there are two key aspects to be examined, those of organizational structure and organizational culture. First it is necessary to look at the structure of
organizations and their originating rationality that is associated with public sector environments. Secondly it is important to have an understanding of organizational cultures, with specific reference to the nature of inherent cultures i.e. bureaucracy - that is associated with such public sector bodies and therefore of particular relevance to this case study research.

This chapter further discusses the relationships that exists between the research themes and sets out the general structure and framework for the data analysis of the case study materials gathered.

6.1 Organizational Structure

There is no one coherent or definitive theory of what constitutes an organization. The most omnipresent model is that of a structure consisting of different functions or departments, which are characterised by a hierarchical composition of precisely defined jobs. Power and authority are designated in line with hierarchical status, and the preciseness of this structure can be represented by an ‘organizational chart’. This chart represents the rationality, it gives order to the purpose of the organization. This representation was formalised upon the ideas of Taylor in early 1900s who put forward Scientific Management as an approach to management thinking (Beynon-Davies 2002).

Scientific Management was one of the first attempts to systematically analyse human behaviour at work. The underlying assumption was that people's actions could be rationalised and defined. Taylor, looking at the interaction of human characteristics, the social environments and physical environments and so forth of complex organizations, drew a parallel to machinery. The machine metaphor explores the parallels between how machines and organizations operate. They
can both be designed to accomplish specific tasks, it presents the image of managers behaving as engineers in order to design and operate efficient organizations in a routine way (Morgan 1986).

The Scientific Management Movement led to growth in formalised middle management structures, and the adoption of rational rules and procedures that are often associated with bureaucracy – a mechanistic approach to organizational activities.

However inconsistencies between proposed organizational activities and actual worker behaviour led to the Hawthorne Studies\textsuperscript{15} (1920s/30s), which identified that workers did not solely respond to classical motivational approaches as suggested in the Scientific Management, but rather their interests also resided in the rewards and punishments of their own work group. Thus it was analysed that an organization is also a social system where focussing on the socio-psychological factors, the attitudes and feelings of workers, is as important as organizing the rational system of formal procedures and rules. Behaviour is often shaped by it’s social contexts (Morgan 1986).

There have been many schools of thought since then and it is recognised that current organizational environments are continuing to evolve. They are becoming more complex and dynamic. Consequently business uncertainty is also increasing such that organizational boundaries are more representative of a variable than a static known entity, which are further influenced by environmental forces such as technology. However as the research case study is a public sector organization and thus highly bureaucratised, it is important to examine what has been

\textsuperscript{15} Hawthorne Studies - Lead by Elton Mayo at Hawthorne Plant of Western Electric Co. Chicago.
happening within the public sector. It is reported that the current trend is to break down the former traditional structures. The UK Government is promoting a Modernization Agenda to reform public services. The key strategy is to achieve reform through the concept of modernization. Central to this is renewal through collaboration, partnership and inclusion. The concept of ‘New Public Management’ is a feature of the Modernization Agenda that is attempting to apply private sector practices to public sector environments. For example, the attempt to introduce markets into particular areas of public sector business and reward people on risk related outcomes (HMSO 1999).

This analysis is particularly relevant to the case study project because it represents a move away from such a traditional risk averse culture to a more risk based environment demanding fast authoritative decision-making. However, in the case study arena problems persisted because the organizational people remained risk averse, adhering to former behaviour and previous work patterns. A key point to recognise is that cultural change seems to lag behind structural change. Therefore, despite changing public sector organizational structures it proves difficult to move over to new cultures where the emphasis focuses on innovation or creativity. Consequently it is important to explore the issue of rationality that underpins such organizational structures.

6.1.1 Rationality

Simon argued back in the 1940s and 1950s that organizations can never be perfectly rational because people do not possess the perfect knowledge needed to make perfectly rational decisions, but rather they work within the constraints and boundaries of the incomplete information of their particular environment. Although in theory the idea is that people adopt a rational model of decision-making that
involves, intelligence, design and choice, in practice individuals are not able to apply rational decision-making because of the lack of perfect knowledge (Beynon-Davies 2002). It is not possible to assess the best means of achieving a goal unless all the information about the means to achieve the end is known. It is necessary to consider all the relevant information and all the alternative options. However this requires unlimited time resources, ‘the number of possible alternatives is so immense that they can’t be examined. The best and only feasible solution is to find a satisfactory solution’ (Frantz 2003, p5).

Therefore decision-making was bounded by the limitations of their knowledge and information. Simon (1976) proposed the concepts of bounded and procedural rationality, and satisficing. He believed that limitations on knowledge and analytical ability influenced people to choose the first option that satisfies their criteria rather than progress through an orderly sequential analysis of the maximum information. He informs that satisfactory decision-making is about outcomes, and that choices are made through subconscious pattern recognition that are borne from experience.

It is important to consider these issues because an organization’s rational decision-making approach does not always match the development approach of the project environment (Morgan 1986).

In organizational literature Weber (1964) determines that formalised organizational structure is associated to the issue of rationality. He suggested that organizational management was best managed using structured and coordinated controls through centralised, hierarchical, and highly departmentalised structures. Thus he theorised that the responses of, and between people could be normalised and organized by managers, rather than simply standardising their business activities.
Similarly Hofstede (2003) applies the term rationality to describe businesses that are seemingly organized to achieve specific goals through explicit procedures. Rationality is culture bound, and depends on the values of the organization. Organizations within a society consist of groups of individuals who collaborate for the achievement of specific goals and a common purpose. Therefore organizational culture and behaviour are formalised around objectively defined areas, and people work towards the clearly established organizational goals. However, Carnall (2003) questions whether effective organizations are rational organizations, i.e. does it follow that a systematic approach leads to a rational result? It is dependent on the definition of rationality, he quotes Weick (1969), who states that rationality ‘is best understood in the eye of the beholder’ (p99).

In reply to this it is recognised that there are many interpretations of rationality applied across a diversity of scenarios, and differing types of organizations, groups of people and individuals. In the context of this research case study rationality is defined as the application of the principles of rationalism to action within frames of reference that are specifically tied to goals of the Department that is bureaucratic by nature. This leans towards a softer version of rationality of the management of organizations rather than an extreme or hard rationality that is associated with science. People do not necessarily base their reasoning on knowledge or experience gained ‘outside the box’, but apply reasoning influenced by the cultural framework imposed through a dominant organizational culture designed to produce efficiency and effectiveness. Put more simply it is an organizational way of doing things within their given frame of reference. In other words organizational rationality comprises of rules and procedures put in place to achieve organizational objectives, which for this research case study are set within a
bureaucratic framework. It is this element that is disturbed when organizational change occurs. Thus, rationality is subjective, different organizations will engender different rationalities due to their differing circumstances, as will their employees. It is relevant now to consider this in terms of a bureaucratic culture.

6.1.2 Bureaucratic Cultures

A structured rationality is endemic within bureaucratic environments that are common to government public sector departments. Weber’s Model of Bureaucracy identified the basic elements of a bureaucratic structure where authority was rational, legal and based on status and position, not on the personalities that are organized in a hierarchy of authority. This view is shared by Hofstede (2003) who states that bureaucracy is a form of organization that is based on strict rules and competencies attached to positions. These views coincide with Wallach’s Organizational Culture Index that profiles a bureaucratic culture as being hierarchical, procedural, structured, ordered, regulated, established, cautious, and power orientated (Wallach 1983). Bureaucratic cultures have clear lines of responsibility and authority, work is highly organized, compartmentalised and systematic (Carnell 2003).

Accordingly, management emphasis is on order, uniformity, and consistency where organizational hierarchies provide channels for decision-making. Bureaucrats make decisions by processing information with reference to predetermined rules. The purpose is to limit the capacity of individuals by exercising control over their activities and decision-making processes. Consequently a linear approach to decision-making is the norm, and one that requires the contribution of other people further up the management hierarchy (Morgan 1986). It provides an effective means for controlling situations that are
fairly certain, but is problematical in uncertain environments (Morgan 1986). This situation is most commonly associated with public sector organizations, and reflects the circumstances in the research case study.

The argument has been that public sector organizations that tend to be bureaucratic by their very nature produce people who tend to be risk averse. They are not prepared to innovate or be creative because they are working within a perceived blame culture. As already stated people are not rewarded on the basis of innovation or ability, because reward is based on the ability to follow the rules. However, being risk averse is related to a premise that public sector organizations utilising public money are accountable to their citizens regarding outcomes and results. Consequently a blame culture is inherent in bureaucratic environments.

Within a bureaucratic setting rationality presupposes an implicit belief that rules and procedures are the basis for order. However, bureaucracy is also associated with red tape where the rules interfere with the achievement of the organizational objectives (Merton 1940). He labels this the 'dysfunctional' consequences of bureaucracy. Therefore it is possible that actions designed to fulfil specific organizational objectives may result in producing counter rational consequences, i.e. unintentional outcomes. It is important to understand, especially when applied to a period of organizational change, that counter-rational behaviour is only dysfunctional to those people who are conditioned to the original objectives, whilst others may regard the behaviour as rational given a different situation, viewpoint or goal (Carnall 2003).

Existing literature exposes a prominent area of debate concerning the effect of the inherent bureaucratic culture that is generally associated with government
projects. In essence, as already discussed organizational culture can be defined as the pattern of basic assumptions that are accepted and used by an organization. Within the context of the case study research, the environment represents a bureaucratic culture. Furthermore as public organizations often operate in non-market conditions, it is often the case that there is no direct link between the services an organization provides and the income it gets for providing them. Controls over these organizations emphasize control over process rather than over outcome, and are therefore culturally based.\(^6\)

6.2 Culture

As referred to above rationality is closely associated with culture, and the researcher believes that for reasons of clarity the term culture needs to be clarified, and the interpretation applied to this research study identified.

Hofstede defines culture as ‘the collective programming of the mind which distinguishes the members of one group or category of people from another’ (2003, p5). He defines two senses of culture, firstly a sense of culture that represents civilization in terms of education, art and literature. This culture he regards as a narrow culture. Secondly a broader interpretation of culture to mean social culture, the science of human societies that includes the more fundamental human processes such as attitudes, feelings and behaviour. It is this second interpretation that is relevant to this research study.

Hofstede’s classic study on national culture characterised four dimensions of cultural differences that are dimensions of values. He defines a dimension as ‘an aspect of a culture that can be measured relative to other cultures’ (Ibid. p14). He

\(^6\) Observation expressed by the Client Project Manager of the Regional Government Department.
explains that a number of phenomena are grouped together to form a dimension because they occur in combination regardless of any logic behind their grouping. This analysis was based on statistical relationships, in other words the tendency of those phenomena to occur in association with each other, not on identified hard links.

Even though there has been some literary criticism about the generalization of Hofstede’s research findings as the sample was drawn from a single multinational company, and participants were consequently subject to a single homogenous organizational culture tied to a specific period (Sondergaard 2001, Goederham and Nordhaug 2001), it is generally accepted as the most comprehensive framework of national cultural values.

A later IRIC\textsuperscript{17} research project conducted by Hofstede across several organizations within one country (the reverse of the previous study), replacing national diversity with an organizational diversity found that at national level cultural differences reside in values, but at organizational level these differences occur mainly in practices. This is because the learning associated with values and practices stems from different social origins. However, this is contrary to other literature on organizational culture that states it is the shared values within an organization that form the core of an organizational culture. In response, Hofstede counters that this is because the literature does not take account of the difference between an organization’s principal values and those of its workforce who shape the organizational culture through their shared practices. It is this latter concept that is relevant to the research case study.

\textsuperscript{17} Institute for Research on Intercultural Co-operation (Hofstede, 1985-87).
6.2.1 Culture and a RAD-type Development Approach

A hypothesis put forward in the literature proposes that factors associated with culture are key to the success or failure of the RAD genre of development approach within large and complex arenas (Jones and King 1998). Protagonists within the IS field discuss culture in terms of inherent organizational cultural constraints, human elements of behaviour and attitude, and the management of people within their organizational settings – these are the main themes of the case study research. For example, cultural factors must be prepared for; and organizational structure, business policies and procedures are put forward as areas of change and risk. Cultural and managerial changes are exposed as two of the greatest barriers to adopting a RAD-type approach (Martin 1991, McConnell 1996). Additionally the risks associated with organizational change are also people oriented, involving their motivation or lack of motivation, changing working cultures such that new behavioural attitudes and work patterns are required (Markus 2004). These views are supported by recent studies (Jones and King 1998, Beynon-Davies et al. 1999) substantiating a need for cultural change, and that culture impacts on the development, the adoption and use of IS.

Thus it is accepted that large projects applying a RAD-type development approach call for cultural characteristics that are different from those used with more traditional development methods. People are required to adapt their work practices and behaviour where the IT system being developed changes the organizational working culture. Research by Jones and King (1998) confirms the importance of recognising the impact of cultural issues. They report on a project where the cultural issues became central to the RAD development approach. Without radical shifts in organizational attitudes and structures, as well as in the mindsets of the
people, many projects fail due to the mismatch between the new methods and practices, and the existing organizational culture (McConnell 1996). For instance, in the research case study it was necessary for the business managers and other business users to move away from previous siloed\textsuperscript{18} methods of working and adopt a more collaborative team approach that involved decision-making skills.

Within the context of the case study it is the aspects of individual, group and organizational cultures that are further discussed because of their relevance to the main themes and development factors under investigation and which are put forward as key areas of impact affecting the research case study.

6.2.2 Individual Culture

People have their own specific thinking patterns and behaviour that stem from their own life knowledge and experiences, and which have been influenced by their own particular social environment. Hofstede (2003) expresses this in terms of a person’s mental programs or mental models, and describes it as their own individual culture. Therefore culture in this sense is learned not inherited. It is different from someone’s human nature or his or her personality. Human nature is inherited and universal to all people. Conversely, personality represents a person’s unique traits that are part inherited and part learned. In other words, personality represents the distinctive characteristics or qualities that belong to one person.

\textsuperscript{18} Siloed - refers to people working together in small discrete teams.
Hofstede uses the analogy ‘software of the mind’ to describe a person’s individual culture. He refers to this as ‘the ‘operating system’ that determines one’s physical and basic psychological functioning’ (2003, p5). It is the ability to feel and express emotions, and to interact with others. He suggests that this genre of emotional activity or behaviour - the human issues, can be influenced by culture. It is important to emphasize that behaviour is not set in stone but only partially predetermined by a person’s mental models. People therefore possess the ability to unlearn previous behaviour, deviate from them and react in new and different ways, creating new mindsets and new behaviours. Thus Hofstede proposes three levels of uniqueness in a person’s mental model, see figure 6.1 below:

![Figure 6.1 Levels of Uniqueness in Human Mental Programming](source)

Consequently individuals working within the project arena as part of the integrated project teams will be subject to their own cultural characteristics but should, in theory, be able to form new mindsets.
As far back as 1987, Willcocks and Mason argued that although human factors are vital to system design, implementation and operation, they are often underestimated and under utilised in new systems development projects. The neglect of human issues in IS development is recognised as a major cause of IS/IT failures and is well documented. They theorised that the way a system is developed impacts on its acceptability and effectiveness. For cultural change to succeed a change in the people's mindsets is needed. It is necessary to convert, but not replace, the existing implicit knowledge owned by the workers with new explicit knowledge of the new culture (Balogun and Hailey 2004). In this way rather than challenging people's ideas, a transition of mindsets is facilitated by the adding of new possibilities and new ideas (Morgan 1986, Willcocks 1997, Carnall 2003 and Hofstede 2003).

6.2.3 Group Culture

In the same way that individuals develop their own characteristics, working within group environments can cultivate group cultures. Group culture is not the same as collective culture that signifies strong unified groups banded together from birth, and which engenders a shared protection of group members in return for unconditional loyalty (Hofstede *Ibid.*). Within the context of the research case study, group culture refers to groups of workers in a shared environment, who having similar responsibilities have built up the same perception of the organization. As a consequence they have developed common mindsets and adopted like attitudes and behaviour. Literature posits that individuals and groups react differently to cultural change. Where an individual is placed in a new working environment s/he will adapt, but where groups of people are positioned into a new cultural setting there is a tendency for them to import their group cultures, i.e. the
culture that has evolved within that group, into the new environment (Hofstede *Ibid.*).

Hofstede (2003) believes that changing such group values in an intended direction is extremely difficult, if not impossible to achieve. However, he argues that values can change, because if people’s working habits are dependent upon the business processes being modified, then those people are influenced by the changes being put in place. He refers to organizational cultures as being Gestalts where the organizational practices are the ‘parts’ able to affect the ‘whole’. Therefore group culture can be influenced by the related organizational structures and systems. However this is only changing people’s habits, it does not necessarily follow that people’s mindsets and attitudes follow suit.

Although Hofstede’s views refer to collective national cultures, there is a parallel to be drawn in the context of the case study group cultures. Business managers, i.e. as a group of workers, allocated to the new development environment imported their former culture with them into the new environment. Problems were experienced in stepping away from their previous culture of deferring decision-making up the management hierarchy, and moving towards the new culture of fast authoritative decision-making. Consequently negative attitudes developed within the new working environment that originated from experiences and bias developed in the previous culture. This is confirmatory of research by Willcocks (1997) that negative attitudes exposed within the new working environment may result from people’s experiences and preconceptions formed in previous work settings.

These concerns refer to people management issues that were influenced by the human issues of attitudes and behaviour. Within a development environment
apposite management is central to change control, to be effective a project must be properly managed, and this involves people management. For this case study people management falls under the umbrella of organizational group culture and concerns how organizational people import their established ways of working into a new working environments. The people management concept considered here does not reflect that of human resource management or project management but to the managing of the project personnel in terms of their supervision, commitment to, and participation within the development arena, and their acceptance of the New IT System. Formerly the responsibility of line managers, people management was subsumed in part by the Department's Project Manager and ultimately the Senior Management Team. Effective people management necessitates a harmony between the social patterns of the organization and the behaviour of the people being managed (Kester 1997).

6.2.4 Organizational Culture

As a concept organizational culture gained popularity during the early 1980s through the work of Deal and Kennedy (1982), and subsequently extended by Peters and Waterman (1982, cited in Hofstede 2003 p179). Organizational culture, also known as corporate culture, refers to the shared mental models of employees in their thinking, feeling and behaviour within their working environment. In other words it is a shared commonality across the organizational workforce. Hofstede (ibid.) defines it as 'the collective programming of the mind which distinguishes the members of one organization from another' (2003, p180). Schneider and Barsoux associate organizational culture with socialization. They state 'Socialization is the process by which new employees absorb the corporate culture and become familiar with the values and behaviour expected of them.' (1997, p135). Beynon-Davies defines culture as 'The set of behaviours expected
in some social group’ (2003, p236), and says that because organizations are seen as social groups they develop an organizational culture. He questions the concept that organizational cultures are organization-wide believing that in reality organizations are composed of many interacting subcultures that develop around structural/functional units or within groups of distinct stakeholders. This was the case with the business managers and business users in the case study project. This point also refers to group culture as previously discussed.

Research by Balogun and Hailey (2004) reports that the strongest barriers to cultural change are the existing organizational cultures. It is often the case that where subcultures are unprepared for change, this is interpreted as resistance to change. As already mentioned behaviour is underpinned by attitudes and beliefs.

Bureaucratic structures require managers to maintain a disciplined and consistent conformity to regulations regardless of the circumstances to which they are applied, such that the conformity becomes a part of the organizational culture itself (Merton Ibid.). In periods of cultural change, the adherence to formalised procedures/processes means that those sticking to the rules then oppose change finding it difficult to adapt to new organizational objectives. Consequently it is the organizational rules themselves that obstruct the new purpose of the organization. As previously stated, these are the unintentional consequences of bureaucracy - sticking to the rules for the rules sake then becomes a means to the end rather than applying them to achieving organizational specific objectives (Carnell Ibid. p102, Merton Ibid.). Organizations with bureaucratic cultures do not generally encourage their employees to think for themselves. Where a strong hierarchical culture is present the resultant structure will be fragmented such that different
functional areas will operate with different pictures of the overall situation. These disjointed views can become barriers to cultural change (Morgan 1986).

Martin stipulates that these types of problems are inevitable, ‘Bureaucracy is the enemy of speed. Indeed it is the enemy of the three main goals of RAD speed, high quality and low cost.’ (1991 p128). Carnell (2003) also confirms that a bureaucratic structure is unsuited to highly complex, dynamic business processes that require a more flexible and adaptable structure and organizational culture. Fixed procedures and static working patterns are not beneficial in a volatile and changing business environment (Crozier 1964 cited in Hughes 2003, p37). Complexity intensifies uncertainty and this in turn increases the need for knowledge in decision-making. Paradoxically there is recognition that complex issues cannot be brought under unilateral control (Crozier 2004). In agreement Highsmith states ‘...trying to build collaborative practices into a rigid hierarchical, control-oriented culture would be folly’ (2002, p128).

Balogun and Hailey (p58) use the analogy of a kaleidoscope to describe the dynamic nature of cultural change. It reflects the constant adaptations that emerge as it progresses. Although some factors are constant they are continually reconfigured to accommodate the difference stages of a change situation.

6.3 The Analysis Framework

As previously discussed, it is the RAD-type development approach that is key to the research study, thus to substantiate that a RAD-type development approach had been applied, the development approach of the research case study was verified against the DSDM 9 principles – a recognised RAD approach (refer to table 5.5, p175).
The main areas of discussion within the literature reviewed were project size and complexity, people management, cultural constraints and human issues. The initial table (see table 6.1 below) further categorised these areas into project factors, people factors and organizational factors, which are suggested to be the most problematical issues for a RAD-type system development approach.

<table>
<thead>
<tr>
<th>PROJECT FACTORS</th>
<th>HUMAN FACTORS</th>
<th>ORGANIZATIONAL FACTORS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of Project</td>
<td>User Involvement</td>
<td>Culture</td>
</tr>
<tr>
<td>- intensive</td>
<td>- active / sustained</td>
<td>- public / private</td>
</tr>
<tr>
<td>- non-intensive</td>
<td>- committed</td>
<td>- nature of structure</td>
</tr>
<tr>
<td>Type of System</td>
<td>Decision-makers</td>
<td>- hierarchical status</td>
</tr>
<tr>
<td>- low complexity</td>
<td>- empowered</td>
<td></td>
</tr>
<tr>
<td>- high complexity</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Size of System</td>
<td>Behaviour</td>
<td>Management /Commitment</td>
</tr>
<tr>
<td>- small</td>
<td>- change receptive</td>
<td>- top management commitment</td>
</tr>
<tr>
<td>- medium</td>
<td>- team working ethic</td>
<td></td>
</tr>
<tr>
<td>- large</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6.1  A Table of Key Issues Affecting a RAD Development

As previously described the factors which impacted significantly upon the development approach were extrapolated, and are referred to as the development factors. These are user involvement, requirements elicitation, user expectation, communications, decision-making and testing. These factors are primarily all associated with cultural issues in terms of organizational, group and individual cultures as explored earlier in this chapter.

For further clarity the researcher presents figure 6.2 below that gives a high level overview of the research journey.
However, it has not been possible to separate either the research themes or the development factors for independent analysis because of the complex relationships that exist between them. It is the development factors that form the structure of the analysis framework as having the most direct impact upon the case study project.

6.3.1 Relationships within the Research Data

In the same way that the complex relationships between the three main research themes and development factors has not facilitated independent analysis, similarly it was not realistic to discuss the 9 DSDM principles in isolation but rather to examine the causal issues through their relationships and connections that affected the ability to achieve the principles of iterative and incremental development and phased delivery of the development project.
A significant factor that was common across the areas analysed involved issues surrounding the effectiveness of communications. Although not one of the main DSDM principles, it is recognised as an integral factor in such a development approach and figured heavily in the research data. Therefore the analysis includes the examination of communications as a development factor, from both internal and external perspectives across the project arena. Additionally it is important to recognise that the project arena was not a closed environment. It was also subject to pressures and problems from outside the development project arena, for example the Foot and Mouth Crisis, and the EC as external stakeholders. Consequently it is necessary to place all the research elements and influencing factors within the context of both external and internal environments. For further clarity the researcher has constructed figure 6.3 that illustrates a holistic view of the relationships described above.

Figure 6.3  A Holistic View of the Factors Affecting the Case Study Project
The above figure illustrates how the RAD-type development approach encompasses the DSDM principles. It shows how these overlap across the three research themes, and the different cultural aspects of people management, cultural constraints and human factors that were influenced by internal issues such as organizational structure and decision-making processes. As can be seen the internal environment was affected by external factors that are specific to the case study project, for example the Foot and Mouth Crisis and the EC as external stakeholders. The researcher believes that clarification of the relationships expressed in figure 6.3 above confirms the suitability of the analysis framework that is structured to present the analysis through the issues that were identified as impacting upon the development project, i.e. user involvement, requirements elicitation, decision-making, user expectation, testing and communications. These areas encompass the principles of the DSDM development approach, traverse across the three research themes and are aligned to the cultural aspects.

6.3.2 Presentation of Analysis

The analysis of the information gathered has generated a vast amount of data that is too voluminous for a single chapter; consequently analysis has been divided into two chapters. It is possible to group the development factors in a number of different ways due to the links between them. It is also true to say that in the literature, requirements elicitation and user expectation are considered to be two sides of the same coin (Martin 1991, McConnell 1996). However within the context of the research case study the connection is less evident because requirements gathering was primarily the concern of business managers rather than the end-users. Whereas end-user expectation and communications had a more plausible cause and effect relationship. Additionally requirements elicitation and decision-making involved the same stakeholders and both issues were also linked to testing
issues. Consequently analysis of the development factors is structured in two parts, firstly Chapter 7 analysis looks at the issues of user involvement, user expectation and communications. Secondly Chapter 8 analysis deals with the issues of requirements elicitation, decision-making and testing. As already mentioned it is all of these areas that affected the development project.

These groupings have been influenced by the natural divide of the data in the context of the research case study. As already discussed user involvement, user expectation and issues surrounding the quality and effectiveness of communications are closely linked. Similarly, requirements elicitation, decision-making and testing issues were interrelated. Although communications have been analysed in the Chapter 7 analysis, it is recognised that issues surrounding the quality and effectiveness of communications also impacted on these areas, and therefore will be discussed in context, but not separately. This is followed by conclusions drawn from analysis in Chapter 9.

The following analysis has been supported by significant remarks drawn from interviews and conversations held with the actual project participants. In this way it is the researcher’s intention to support the analysis and conclusions made by drawing from the rich data gathered to further illustrate and substantiate the points being put forward, rather than to use them directly as proof of analysis. To maintain anonymity people are given a numeric identifier and a job title, such that all comments made by Business Manager (5) reflect contributions by the same person. However where the style or content of the comment presented enables identification of particular individuals the number has been replaced with an alphabetical reference e.g. Business Manager (b). In this way it has been possible
to maintain a consistency and continuity within the validation process, and to protect the confidentiality and anonymity of the participants.

6.4 Summary

In summary, this chapter has defined and distinguished between the main research themes and the development factors and their association with the DSDM principles. It has set out and examined the forms of culture relevant to the context of the case study project in terms of individual, group and corporate cultures that reflect the people management, cultural constraints and human factors. It further examined the relationships between these themes. Finally it presents the general framework for the data analysis that involves the development factors of user involvement, requirements elicitation, decision-making, user expectation, testing and communications that is separated into two chapters that divides analysis into two parts, Chapter 7 focuses on user involvement, user expectation and communications. Chapter 8 concentrates on requirements elicitation, decision-making and testing issues.
7. Introduction

This chapter sets out the analysis of the case study data gathered. The analysis framework of the three main research themes, the development factors, the DSDM principles, and the cultural influences together with their associated relationships as examined in Chapter 5. As previously explained presentation of the analysis has been divided into two chapters.

Chapter 7 analysis concentrates on the development factors of user involvement, user expectation and communications that affected the development project. This grouping has been influenced by the natural divide of the data in the context of the research case study. User involvement was closely linked to user expectation and, issues surrounding the quality and effectiveness of communications impacted upon both areas.

More specifically the Chapter 7 analysis focuses on the importance of selecting the right users, their availability and, subsequently their allocation to the project, also taking into account the degree that these issues were affected by external factors. Further analysis of user expectation examines levels of expectation and its management, both of which had a knock on effect to user acceptance that was also linked to training issues and the lack of prototyping.
Finally, the analysis turns its attention to communications, their quality and efficacy. Both internal and external communications are considered in relation to the bureaucratic nature of the development environment, through the exploration of tacit, explicit and implicit knowledge. User involvement in relation to requirements elicitation and decision-making is explored in the Chapter 8 analysis and its sections. Preliminary conclusions representing the impacts and the affect that each development factor had upon the RAD-type development approach are presented at the end of each section. Final conclusions are put forward in Chapter 9.

7.1 User Involvement

As discussed previously the term 'user involvement', which is the first of the 9 DSDM principles, is used to reflect the concept of user engagement that combines both user involvement and user participation as discussed in Chapter 4, Section 4.1.2, p126. It relates to the indirect business users i.e. the Scheme and Process Managers, other business managers and CAPM business people situated in the project environment, and the direct users i.e. the end-users located in the Divisional Offices (DOs). It is the former group who were primarily involved; the latter groups are more associated with issues surrounding user expectation and communications. Where the term user community has been used this refers to a coalescence of all the groups. A high degree of end-user involvement, a key goal of a RAD-type approach, is accepted as being crucial to meeting user’s needs.

7.1.1 Right Users

User involvement in systems development is a complex issue. It is multi-faceted. There are many issues within this case study research of how it has impacted upon the development approach. As previously mentioned there is much discussion surrounding the identification of the right users. Hirschheim (1983) and
Beynon-Davies et al. (1998) support the view that for large and complex development environments, where it is not practicable to involve all system users, the issue of identifying the right users is crucial to system development.

In analysis the researcher believes an important point to argue is that due to the innovation surrounding the Generic Process, workflow and business rules concepts it would have been difficult to identify individuals from the end-user community who would have been knowledgeable in these areas. It required individuals to have the ability to design the screens associated with the new business processes. Consequently this role was allocated to the higher level indirect users who, it was assumed, had knowledge of both the Generic Process and the agricultural context that included the EC legislation. Although literature emphasizes the involvement of end-users this proved difficult, the view is that they lacked the expertise and commitment, for example:

... we tried at the beginning to get the end-users involved and partly because of the locations of the end-users, partly because their lack of expertise in what we were asking them to do. it didn't work ... it didn't work out and because of lack of time, because we were so late, they were so late delivering things ... it would be the middle people saying well yes they could live with this if they really had to, where it wasn't the end-users and I agree that we should do far better than we have for our end-users. Business Manager (3)

The Suppliers felt strongly about the lack of end-user involvement and hence the benefits that could have been derived from their feedback, i.e. feedback from the actual system users. Supplier comments to support this are set out below:

One of the unusual aspects of this has been that we don't really have any involvement with end-users as such. The requirements that we are given, are given to us by process managers. Occasionally we've set up workshops with process managers and the process manager may bring along somebody from DO but it is fairly rare. Supplier Developer (3)\(^\text{19}\)

\(^{19}\) This quote has also been used on p279 to illustrate a related point.
... we are definitely not working with people at the front end, because as well as improving the system that would have a benefit of being the mechanism for providing some additional positive feedback to the Business. Supplier Developer (8)

7.1.1.1 Right Users – Inflexibility of Business Analysts

It is recognised that a fundamental principle with a RAD-type development approach concerns the structure of the project teams involved, teams are particularly important. There is a link here to the issue of identifying the ‘right’ users for a project team, previously voiced by Ljubic and Stefancic (1994), that people who have an ‘arrogant’ attitude should not be placed in RAD development teams. In the context of the research case study the arrogance referred to relates to the uncompromising attitudes of some Supplier's developers during the initial stages of development and their insistence on adhering to a pre-defined scope. However the difficulties with scope also caused concern for the other Supplier developers who also found it difficult to negotiate with users, as they also felt restricted by the pre-existing development parameters. A Supplier developer involved at that time confirmed this view and illustrated it through the following comments:

... for the first 8/9 months I spent most of my time telling the Business 'sorry that's not in my scope', because I had a very strict list of requirements that I had try and deliver and anything outside of that was change control and had to be managed by the Project ... Supplier Developer (6)

I know for a long time that I kept having to tell the Business 'no that's not in scope' which frustrated them enormously. Supplier Developer (6 – second interview)

These comments exemplify the view expressed by McConnell (1996) that without a common view, identity or trust project teams will fail. This situation is also illustrative of views put forward by Balogun and Hailey (2004), who believe that
where there is a high degree of change in project development a common vision is required to prevent differing or individual interpretations of the project goals in order for the workforce to gain a common understanding of a shared vision. The developers and business people involved at that time had differing views as to system scope, and ultimately of the business requirements.

Low levels of commitment to the project compounded this situation. For this case study there was recognition by the senior management that there were difficulties assigning not only the correct users, but also obtaining the required level of commitment for the project. A senior manager made the following comments:

*I think that there was a tension all the way through this, we struggled to get the right Business people onto the project right from the onset and I think to get the right commitment.*  Senior Manager (3)

*Senior management had a commitment for what we were doing and for what we were trying to achieve, I think lower down in the organization there was scepticism about what we were trying to do and therefore there wasn't the support so a lot of lip service was paid to it and we struggled to get the right people.*  Senior Manager (3)

Evidence suggests that within the initial workshop environments that took place during the first year the concept of collaborative working proved challenging. The Suppliers, who had been provided with the Generic Process concept at procurement, held a number of design JAD workshops involving their own business analysts and the Departments’ business managers who represented the business needs. However, there is evidence to suggest that some of the initial Supplier’s analysts involved were working to different objectives than those of the business managers involved. These Supplier’s analysts believed that their commitment was to design and develop to the initial remit that they had been given, which defined their development scope and their parameters. The business managers on the other hand, were committed to a design that would meet all their
business needs, and this went far beyond the high-level Generic Process requirements referred to. Hence a view held by the business managers was that the Suppliers were confined to a remit that was pre-scoped to the high-level requirements of the Generic Process presented at procurement, and which did not meet their specific business needs. This difference in viewpoints was the cause of some conflict between the business managers and Supplier's analysts early on in the project.

The researcher discovered through discourse with the two groups of people concerned that rather than creating an integrated project milieu, there was a perception of two sides, 'us' and 'them', each party would make references to 'us' or 'them', for example:

... I think everyone is working well together now but initially I think there was an 'us and them' attitude. Developer (8)

Initially yes, it was very 'them and us' Business Manager (5)

However it should be recognised that these difficulties between the parties arose during the initial stages of the project. As new Supplier people were brought on board during the first personnel changeover, there was a move towards a more flexible project attitude. The project environment switched from being perceived as inflexible and contentious, to one of change acceptance that evidence posits was partly management driven and partly due to the change in project personnel. Management decisions were made behind closed doors; it is therefore not possible for the researcher to define more precisely the reasons that influenced the acceptance of change in project scope other than to suggest that there was some amendment to the commercial negotiations that enabled the Supplier developers
to behave more flexibly towards the business requirements. Interviewee comments in support of this theory are set out below:

*For some reason it then seemed to switch into an environment where OK it's a change but we'll take it on board, we'll raise the change control ... we'll just be here a bit longer.* Supplier Developer (13)

*The fact the Business keep throwing new requirements at us and I had to deflect them for a time, then changed to that's OK we can take them on.* Supplier Developer (6)

*... there were some key individuals in the first part of the project and if the people that were in post now ....... had been in place then we'd have had more of what we wanted and a more workable solution than what we were given.* Business Manager (9)

The researcher confirmed that there was some revision of Supplier's analysts during the first personnel changeover early in stage 2 of the project - see Section 5.1.5 (p149), which details the periods of change that occurred and which are referred to throughout the analysis.

It is not clear whether this was due to the identified problems of scope, or if new people were brought on board as the project environment evolved. Analysis puts both viewpoints forward. Nevertheless it was at this time that a more flexible attitude to project scope in terms of design, development and system change requests was adopted by the Suppliers, aided by the in-coming personnel.

Additionally a more people focused management approach was adopted by the Suppliers towards the business people, which combined with a more holistic cross scheme development approach, engendered a new attitude towards prioritization of development work and a more collaborative working environment, that is illustrated through the following comment:
...people then started to give and there was a marked difference in some of the meetings where they were actually taking practical decisions and saying 'that can be left, this is more important'. Suppliers Developer (21)

This initial conflict is an interesting issue that also affected requirements elicitation and decision-making which are covered in the Chapter 8 analysis. The wrong blend of people prevented team fusion, leading to inaccurate communications and hence development problems – these issues are discussed more fully in Section 8.1, p272.

7.1.1.2 Right Users – Exclusion of Former IT Personnel

Historically, within the Department a separate IT branch dealt with CAP scheme changes. The former IT branch of CAPM consisted of 120 Operations Support staff, and extended across the three DOs who had the responsibility for maintaining the CAPM IT systems. In the past the old IT department was considered to be too powerful. The perception was that they made business decisions at an IT level, and had a perceived reputation for delivering programs late, that also did not always meet the needs of the users. More recently, however there is evidence of some successful IT development and delivery, but in spite of this there was a definite decision by Senior Management not to involve the former IT department in the new system development.

Evidence interprets the exclusion of the former Department's IT personnel from the project environment as a significant contributory factor to initial project problems. The decision that they would not be involved was a senior management one and made at inception. When questioned senior managers put forward a number of reasons to justify their decision. Firstly, that the existing IT resources were needed to maintain the Legacy System and implement any changes to
support the day-to-day business activities. Secondly, senior management felt that there was resentment towards the new system from the IT people who were fearful of losing their jobs despite reassurances to the contrary. Thirdly the negative attitude held would not be beneficial to the project environment itself.

Thus, those IT individuals who possessed crucial inherent knowledge, an understanding of the difficulties involved with the schemes complexities, who also had experience of the interwoven relationships and, domain specific IT development issues were not utilised. Their potential contribution and benefit were lost. Consequently, initially too much faith was placed in the Supplier’s personnel who, it was assumed, held the necessary IT expertise, and had sufficient understanding of the agricultural field and subject specific business requirements. This proved not to be the case as illustrated in problems experienced with requirements elicitation (see Chapter 8 analysis, Section 8.1, p272). Through interviews with former IT personnel the researcher was able to establish this was an accurate analysis of this issue, and confirmed the situation with senior management, and puts forward the following quotes in support of the above analysis:

_They (Suppliers) came in here and said this is what you want so there is no point asking any of the old IT people. There is a lot of experience out there that know what the pitfalls are. I think if they have been there from the start they would have picked up a lot of the problems, they’d have identified a lot of issues that have created problems._ Business Manager (5)

_... some of the IT people as well, in hindsight we should have pushed harder to get more people. We pushed hard but we lived with what we had in the end and it wasn’t enough and it wasn’t of the right quality and that was unfortunate._ Senior Manager (1)

In analysis the researcher believes that the decision to exclude the IT experts from the development arena was flawed.
7.1.1.3 Right Users – Testing Issues

For this particular case study there was a heavy dependence on the UAT team\textsuperscript{20} that provided a mechanism for feedback. This section looks at the issue of right users in terms of the problems experienced with project activities with particular reference to testing activities. More specifically this section asks whether those users performing testing were the right ones. Integral to the issue of selecting the right users was the type of users involved particularly in relation to the UAT testing stages of the project. As stated above previously users reflected two distinct groups, (1) the indirect business users i.e. the Scheme and Process Managers, business managers and general business staff situated in the project environment and (2), the direct users i.e. the end-users located in the DOs.

However specific problems relating to the test environment that impacted upon the development project are the subject of analysis in the Chapter 8 analysis, Section 8.3, p297. Similarly issues specific to requirements elicitation are explored in Section 8.1, p272.

Evidence reveals that the business managers, i.e. Scheme and Process Managers, specifying the requirements against the EC legislation, were not always the same people actually testing that those requirements had been met. Data gathered reveals this was performed by the User Acceptance Test team (UAT). The UAT team consisted of former DO staff relocated on the project development site. Using their specific scheme expertise, they worked using test scenarios put forward by business managers, from which they designed test scripts to determine whether the requirements had been achieved. Literature (Martin 1991, Beynon-Davies 2000, McConnell 1996) advises that those people involved in testing

\textsuperscript{20} User Acceptance Test Team.
whether requirements have been met should be those defining the business needs. However in the context of the research case study in the initial development stages, although it was the business managers who were involved in requirements gathering, they were not always directly involved in ascertaining if their requirements had been met, analysis suggests they essentially verified and signed off the testing performed by the UAT team.

An associated risk linked to this, raised by the Supplier's developers, was the question of whether the scheme specific knowledge holders of the UAT team had adequate knowledge of the both Generic Process and of the different CAP scheme requirements to be able to perform rigorous testing. The risk here was whether the testers would know if the requirements they were testing were the right ones. For example, if a function did not appear on a screen interface would they have sufficient knowledge to appreciate that it was missing, or that it provided the correct usability cross the CAP schemes. Analysis posits that had the former IT personnel been involved their knowledge and expertise would have been of significant benefit in identifying both usability and functional requirements.

A further concern, expressed by the business managers was that the design, build, test cycles that effectively replaced the prototyping element of the development approach due to time constraints, meant that UAT was their first opportunity to view development releases. Consequently if a development 'item' was not ‘fit for purpose’ then it required more development effort to amend it than it would have needed if viewed in a prototyping stage. Additionally if it was seen as ‘fit for purpose’ this was often achieved at the expense of the look and feel elements. A Business Manager spoke about his concerns on this matter, and further research

CAP = Common Agricultural Policy Schemes.

21
confirmed that his views were supported by other business managers. The following interviewee comments are representative of the views across both parties:

... the first I get to see anything is in UAT and then we are at the stage there where if it's not 'fit for purpose' then it has to go back, but if it is 'fit for purpose' regardless of how aesthetically pleasing it is get it done. Business Manager (6)

I get the feeling that the people who are doing the business testing, the UAT don't have that much of an understanding, although they understand the business, they don't understand the system we are implementing. Supplier Developer (18)

Traditionally UAT is performed to ascertain the usability of the system rather than test the system itself. This is an important issue, it deals with integrated testing, DSDM Principle no. 8, and is further discussed in the Chapter 8 analysis.

With hindsight it has been accepted that the people doing the testing should be the people who set the requirements and that UAT should be confirming the usability of the system. Therefore if it is the case that business managers are specifying requirements, then it is necessary that they then perform the testing activities themselves.

One of the things that we've learnt ... is the need for their [Business Managers] involvement during analysis and during user acceptance testing, they've realised the benefits of doing so [themselves] rather than just throwing it over to the UAT team. Suppliers Project Manager (2)

7.1.2 Availability of Users

Putting the above issues to one side, another aspect that affected the ability to deliver to the 5th DSDM principle, the iterative and incremental elements of the IAD process, was the actual availability of the key Client staff. Formal project
documentation from the planning stages implies that dedicated business people empowered to make authoritative decisions, together with an end-user team would be allocated to the project to participate in prototyping and, who would be able to make timely decisions such that the iterative process could occur as planned. This did not happen and analysis of data gathered puts forward a number of reasons that reflect this.

Culturally bureaucratic environments with siloed\textsuperscript{22} patterns of working operate on a 'one person, one job' basis. Thus specific skills sets and business knowledge belong to individuals making it difficult to back fill posts when people were allocated to the project. As a consequence the domain specific tacit knowledge owned by individuals was not shared or passed on. It is important to also include the exclusion of the former internal IT people who possessed both the domain specific knowledge and the IT expertise meant that their wealth of tacit knowledge was lost to the project.

Additionally the need to maintain the day-to-day, Business as Usual activities meant that business managers were reluctant to commit to the project. Consequently their availability was limited and their commitment partial. Analysis thus confirms previous research by Beynon-Davies \textit{et al.} (1998), it verifies that within the context of this research case study different types of users, held different types of knowledge and were subject to differing levels of availability. It was often those people maintaining the day-to-day activities, who were the least likely to be available, but who were the key knowledge contributors. The Client Project Manager commented:

\textsuperscript{22} Siloed - refers to people working together in small discrete teams.
I think we had this gap ... and really the business expertise at that time, because at the time we needed them they were all off doing other things to keep the organization running. So we suffered really not having the right expertise available to us and, necessarily that expertise has to be available on demand. That is the conflict, people have their jobs to do. Client Project Manager (1)

People became resigned to long hours and overtime in an effort to meet deadlines. Analysis shows that at the time when delays were at their worst, during year two of the project, it was common for the Supplier’s developers and the business managers to work over weekends. For example one Business Manager reported:

I for one, from Christmas from 2002 to July 2003 I don’t think I had a weekend off, not one. Business Manager (A)

The situation arose due to the former ad hoc development culture of the CAP schemes. People were allocated to scheme responsibilities discretely such that knowledge was retained individually. Evidence reveals that this concern was recognised across the project community, but proved difficult to resolve. There was a general apprehension about the situation. It presented a real risk of project failure, ‘what if’ one of the key business knowledge holders became inoperative? Consequently it became necessary to widen the knowledge group, to ensure the availability of the business expertise that was specifically crucial to the Generic Process. When questioned on this matter a senior manager’s reply confirmed this analysis:

I think what we failed to do over time is to actually widen that knowledge to a wider group of people and that counted quite a lot when you started then getting into areas of Generic Process, getting people to think more generically, that was quite difficult and still is quite difficult. Senior Manager (4)

Other comments gathered from the project community that confirm this view are set out below:

23 To prevent identification such that other comments similarly labelled can be recognised and attributed to one individual the identification number has alphabetized.
... the difficulty is that the knowledge is held within a few particular people, and they are the people that are called upon all the time, they are the busy ones, they are the key people ... they are the experts in their field and if one of those is removed then the Project would be in trouble. CAPM Business (2)

God forbid if someone had been run over by a bus, the whole system would have really gone back ... Business Manager (1)

Because there is only one person that knows this' they have to be in 'that' meeting but they also have to be in another meeting at the same time in a different place. Business Manager (1)

7.1.3 Allocation to the Project

A major external factor also had a part to play in obstructing user involvement in the iterative and incremental elements of the project development cycle. An outbreak of the Foot and Mouth Disease affecting the agricultural industry during the first 6 months of the project impeded development work considerably. Both key business managers and end-users left the project environment and were re-assigned to deal with the crisis. As a result the unavailability of end-users with the detailed user knowledge meant that those users who were involved in the initial JAD workshops were not from the user end of the environment, for example:

We lost a lot of staff to work in different locations, the staff that were left were inundated with day-to-day work for the staff who had left for Foot and Mouth couldn't do, and so I think the impact on the project was it could have used greater leadership skills within the Division staff to its benefit. Business Manager (6)

Added to this was the geographical location of the end-user groups. A significant number of them were located a great distance away from the centralised development environment. These people were not enthusiastic to take on board the necessary travelling, or to commit to working away from home during the week as the following comment demonstrates:

24 Details of the Foot and Mouth crisis are set out in the case study. Issues surrounding the impact of the Foot and Mouth crisis are more fully discussed in Section 7.1.5, p233.
.... when we tried to get a cross section involved, we faced issues about travelling, we faced issues about their managers not being willing to release them, we faced issues along the lines of 'well I can attend but I can't take any decisions only my manager can make the decisions. Developer (8)

There is a link here with cultural issues inherent within bureaucratic hierarchies, and to organizational structures and decision-making processes that are described in Chapter 5, and fully discussed in Section 8.2, p289, of the Chapter 8 analysis.

7.1.4. User Involvement – A Management View

The researcher asked members of the Senior Management Team to reflect upon problems associated with user involvement and, consider retrospectively what solutions could have been applied. The following comments are representative of those made:

*I think we would have pushed a lot harder for that (the involvement of former IT people) to have happened and tried to have found a way of keeping the existing business going in some other way to release people to us.* Senior Manager (1)

*In retrospect ... involve the Business more at the outset, make sure that we have the resources there that we can actually free up rather than have to ask people to do things as a favour or in addition to their day job because that's never easy when it becomes out of control.* Senior Manager (2)

*It would have been more appropriate to have more professional resources around us during the earlier stages of the project. We took a lot of people who were new to project, who weren't really familiar with it ... I would want more support around me.*

*Does this include the former IT people who were excluded?*25

Yes, you would want them right in the centre of the project. Client Project Manager.

The above comments to some extent sum up the issues discussed above. There was senior management recognition of the on-going problems being experienced but little evidence of proactive leadership. The development environment became

---

25 Bold type indicates follow on question posed by the researcher.
a reactive arena where problems were experienced and then if not resolved, addressed in an effort to maintain a level of project activity rather than aggressively progress it forward.

7.1.5 User involvement – External Factors

A major external factor implicated with user involvement is that of the Foot and Mouth crisis that occurred early on in the project. As the reader has already read, the crisis impacted upon all the areas discussed above, and was therefore an important contributory factor that needs to be analysed. Analysis reveals that there was a dichotomy of views over the crisis. There are two schools of thought that are represented across the project community. Firstly, that the Foot and Mouth crisis had a significant impact on the iterative development approach, and secondly that it was only a marginal causal factor in the project delays experienced. It is interesting to explore this debate further through the data gathered from interviews held with the Senior Management Team, the Suppliers and the business managers that effectively demonstrate this dichotomy. Thus the discussion unfolds firstly through opinions that support the view that the Foot and Mouth crisis did have a clear impact on the project and secondly by presenting opinions to the contrary. The views put forward are then analysed further. To begin therefore the following comments present the case representing a significant impact:

I'm clear in my own mind that Foot and Mouth did have a significant effect on the early stages of the project for several reasons, first of all, in terms of the senior staff who were holding the reins and keeping an eye on the project, I for one who was the Chair of the Project Board, was taken away for 6 months. ... Senior Manager (4)

People involved in managing Foot and Mouth were probably the people we needed for the early IAD design and even if we were able to get hold of them their time was not dedicated to the Project which in some cases they should have been to make the Project successful, but that was something that nobody could legislate for. Supplier Project Manager (1)
It had an enormous effect I think on staff availability on what probably should have been, would have been the Businesses highest priority i.e. implementing Programme of Change suddenly being pushed to one side and the priority being somewhere else. At the highest levels it meant that the senior management (Team) were either distracted or formally seconded from their normal jobs to do other things. Supplier Developer (8)

Now consider comments that reflect an opposing view that the Foot and Mouth crisis only had a marginal impact:

I don't think that there was a major impact on the IT side of things. The major people they needed to speak to were the Process Managers and none of them got involved in the Foot and Mouth situation ... but from the Business side where I was working was carrying on with the project and I saw no change at that period. Business Manager (3)

As far as I can see there has been virtually no impact on this project. It is being used very well as a reason for delays ... but I would say very little impact on this project, and that it was not a valid excuse. Business Manager (8)

... we weren't involved in the Foot and Mouth crisis in any way. The only effect that Foot and Mouth crisis had on us was having to have extra controls and what have you in place... but the amount of involvement we had with Foot and Mouth crisis was minimal compared to a more fundamental problem of not working with them (Suppliers) from the start. Business Manager (5)

In analysis, from the Senior Management Team point of view, the impact of the Foot and Mouth crisis had a serious impact upon the iterative development approach affecting the project in domino fashion as project delays cascaded down the line. They believe that focus and control were removed from the project as the senior managers were reassigned, and that the business managers with crucial business knowledge were also either reassigned or unavailable due to pressure of trying to accommodate their own Business as Usual activities, and those of personnel reallocated. This situation also affected the early JAD design workshops such that the detailed user knowledge and involvement necessary was unavailable as those present were not from the user end of the environment. There is here a tenable link to requirements elicitation that is analysed in Chapter 8.
The Suppliers put forward the same perception. From their standpoint the Foot and Mouth crisis had a critical effect on the project such that the crisis became the prime priority, thus removing focus from the project. It is their opinion that the project suffered from the lack of early participation of the critical business users such that design and development were delayed. They posit that business motivation and commitment were reduced at a time when the foundations of the project were being created. They also believe that the damage occasioned to the IAD development process in the early stages had long-term implications throughout development.

From a business manager’s perspective however, a different view is presented. The majority report that they saw little impact of the Foot and Mouth crisis on their work. They maintain that those with the crucial business knowledge were still available on site, that the people who were removed were the senior managers, and that those employees removed to deal with it from the business community were the day-to-day operators who were not involved with the business end of the development environment. Contrarily there were two business managers who believed that the impact should not be underestimated. They posit that a particular issue was the lack of detailed end-user knowledge at the initial JAD workshops, which is an issue that has been highlighted previously. They further state that although they were around during that time their availability was restricted, as they had to maintain the Business as Usual activities mentioned above and deal with work from reallocated colleagues.

Thus it could be analysed that the impact of the Foot and Mouth crisis may have been dependent on the degree of individual involvement in the development process at that time, and/or also due to one’s perception of the project. The
researcher did not start on the research project until after the crisis and is therefore unable to contribute any personal observations or evidence. However a conclusion that could be drawn is that at the time of the Foot and Mouth crisis the design and development areas being attended to reflect the areas of responsibility of the two business managers who reported that the crisis had had a significant impact on the development project.

7.1.6 User involvement – Conclusions

Although there are many variables within development arenas that affect implementation, the areas analysed above are those which were significant to this research case study. It was the negative feelings, and feelings of uncertainty formed during the early stages of the project that prevented the mindsets of the user community from moving on. The importance of engaging key stakeholders i.e. the user community, during periods of change cannot be denied, this is confirmatory evidence of views expressed by Carnall (2003) that such stakeholders need to be involved and kept informed.

The initial inflexibility of, and adherence to an assumed project scope coupled with the conflict that arose between the business managers and the Supplier’s personnel did not promote the collaborative working environment that is needed for a RAD-type development approach. Problems experienced with the availability and allocation of the right users constrained by Business as Usual activities was compounded by the exclusion of the former IT personnel which meant the crucial business knowledge was not readily available. However a valid argument reflects the innovative nature of the system being developed meant that the end-users would not have possessed the type, and level of knowledge needed for the
development process, consequently it was the indirect, higher level users who were involved.

The initial low levels of key knowledge holders and end-user involvement meant that development did not get into the powerful iterative cycles that had been planned. It was replaced with incremental delivery cycles reflecting design, build and test stages, rather than the early visibility feedback of the planned iterations. As a consequence system test and UAT\(^\text{26}\) became the mechanisms for change and problem resolution. Although this is a factor that project management could not have planned for, it could nevertheless be argued that because the Suppliers were working on the high-level requirements specification to produce the Generic Process, specific input from the end-users was not crucial. Therefore analysis suggests that although contributory, this issue was not solely responsible for the lack of development iterations.

The restricted user input into the early stages of project development meant that the system scope was inaccurately defined, and that inaccuracies experienced during user testing served to reduce user confidence of, and commitment to the new system. This analysis confirms previous research by Jones and King (1998) who report on a failed RAD project where the culture only permitted part time commitment from the user community. There was an assumption that once the right users had been selected, these people would make themselves available, and routinely adopt a high-level of commitment – this was not the case. The mismatch in the initial requirements gathering was not wholly a result of the development approach applied but due also to cultural differences between developers and organizational people.

\(^{26}\) UAT = User Acceptance Test team, this is fully analysed in Chapter 7 analysis.
In general the Foot and Mouth crisis is perceived to have had considerable impact during the initial stages of the project that impacted down the development line. A sceptical opinion is that the crisis only had a marginal causal impact. A more radical view that is thought, but not openly voiced, is that the crisis had a real but limited impact upon the project but that it was a convenient vehicle through which to justify the severe project delays experienced. Indeed in the National Audit Office Report (2003), the Foot and Mouth crisis is put forward as a main contributor to project delays. The researcher is unable to satisfy this comment either way and therefore leaves it up to the reader to draw his or her own conclusion.

7.2 User Expectation

Literature expresses the view that expectation management is a major success factor in a RAD-type development approach. Expectation failure is defined as the inability of an information system to meet a specific stakeholder group’s expectations and can be linked to JAD workshops during the initial stages of project development.27

Within the context of the case study where user expectation was unrealistically high, a number of factors were involved. As previously explored there were essentially two groups of users, the business managers (indirect users) and the end-users (direct users). On one level some business managers considered their expectation levels to be related to the degree of their involvement in the JAD workshop activities. Although user expectation of the end-user community is linked to user acceptance that is discussed in this section, it is also concerned with communication issues that are more fully covered in Section 7.3, p254.

7.2.1 User Expectations - Unrealistic Levels

Senior management accepts that user expectation of the New IT System was unrealistically high. Analysis posits that low levels of end-user involvement as discussed above, together with poor and ineffective communications throughout development were contributory factors. The initial expectation for the New IT System is well documented, but it has not been realised either in terms of the actual system itself or delivery dates. The image of the system projected by the Suppliers and the Department was that of a panacea, a cure for all the old problems. It would remove most of the manual work, deal with 'perfect' claims automatically without manual intervention and, improve the work experience of CAPM staff. When they defined their business needs the business managers specified everything required to fulfil their business responsibilities. The attitude was that they expected to get all the initial requirements they had requested. One Business Manager commented:

... 'I was told this was what I was going to get therefore it's my job to make sure that we get it' Business Manager (14)

This view was never realistically challenged and is more representative of a traditional up-front development approach, than a RAD-type application employing the use of timeboxing\(^2\) that is crucial to this genre of approach. It is an important issue and needs to be drawn out. If timeboxing is used to manage user expectation there has to be clear consensus between all parties of what is being developed. In the context of the research case study the timeboxing concept was not applied effectively such that a disparity between the views of the Supplier’s developers and

---

\(^{2}\) Timeboxing, an element of RAD, is explained in Section 3.5.4.2, p95, and analysed further in Chapter 8 analysis.
the business managers led to unrealistically high levels of expectation, and hence expectation failure.

Consequently issues with signing off development work occurred because business managers did not feel they had been provided with all the business functionality they had asked for. What were delivered were the core requirements across the Generic Process, the ‘Must Haves’ rather than the ‘Nice to Haves’ as referred to in the MoSCoW\(^2\) element of a RAD-type approach.

Set out below are some examples of the overly optimistic vision that was put forward by senior management to the user community: The researcher offers 5 interviewee responses as supporting material to demonstrate the strength of feeling behind this issue rather than limit user representation to 1 or 2 comments:

... I don’t think we ever spelt out to them or worked with them to manage what we might have been able to deliver other than what we were saying ‘this is an all singing, all dancing ... and so on type of investment’. Senior Manager (2)

... I think that the expectations were very, very high even from the initial presentation of when the Suppliers were awarded the contract we have this big ‘wizzy’ presentation ... so everybody’s expectations were really high, we were all expecting the Rolls Royce and at the end of the day we had a little 2CV. Business Manager (9)

I think they (Suppliers) did paint a very rosy picture of what was going to be available in a relatively short period of time and I think that maybe we exaggerated the expectation I think. Business Manager (6)

... it’s been evangelistic. This is the greatest thing since sliced bread. They (Management) really sold this hard. It was made out to be an incredible system but it’s not what they’ve got. Business Manager (4)

I think we may have over sold that a little bit in terms of expectations of people. You know, we’re going to set this up, you’ll be able to see it, look, feel and touch it ... that didn’t really come to fruition. Senior Manager (1)

\(^2\) The MoSCoW aspect of RAD is explained and discussed in Chapter 3, Section 3.5.4.2, p95.
These views are representative of the user community, and although expressed during the early stages of the project, they were still put forward as an example of what was expected 2 years later. Analysis suggests that those indirect users who were more closely involved in the system design and development failed to disseminate project progress in a manner that challenged this perception to the direct users (see Section 7.2.2 below). However this is not a straightforward issue, it concerns effective communications and the distribution and exchange of such material. These issues are linked to user perception discussed next and to communications that is further explored in Section 7.3, p254

7.2.2 User Expectation – User Perceptions

In addition to false expectations of the system, end-users located in the DOs30 were not kept informed of delivery dates such that when delays occurred and delivery dates slipped they were not always made aware of changes in the development / delivery schedules. Their initial belief was that one day they would cease working with the Legacy System and move on to the New IT System. One view is that because the DOs were physically located away from the development environment they felt left out, and that this influenced their perception of the system. Reactive recognition of high expectation levels prompted key persons to visit individual DOs to converse with the end-users and ascertain what was expected. However as far as the end-users were concerned this was unproductive, there is no evidence that any feedback gathered was taken on board, and none of the suggestions put forward by distant users were dealt with. Analysis here emphasizes that this was not effective management of user expectation. Interviewee comments in support of this are set out below:

30 Divisional Offices.
Oh yes, our expectations have been mis-managed, we thought on the 1st January they'd flick a switch and New IT System would be fine. CAPM Business (17)

I remember, it was going to be released this afternoon, it was 6 weeks before it was released yet it was supposed to be that day. We (the DOs) were expecting it ... I'd much prefer them to say 'we don't know, it's not going to be in the next month at least' and we'd understand. CAPM Business (3)

The perceived mismanagement of user expectations of the business managers caused problems for the Supplier's developers working to tight deadlines. Discord resulted from their descoping business requirements to achieve a 'fit for purpose' concept rather than comply with the range of requirements requested by business managers. Analysis confirms that even those working closely with the development team were expecting to get the requirements they asked for. When questioned senior management reported that they had in fact identified, but not resolved the realization that a more 'hands on' approach was necessary to educate and inform the user community. Comments to support this are set out below:

One of the things we've had to overcome is the perception of the Business because when we came in originally and said 'this system is going to do all this and it is going to do it by XXX', ...from then on you are fighting a very uphill battle with the actual user community and the Business who say 'well you said you were going to deliver this and it's not been done'. Supplier Developer (21)

... we were still expecting them (Suppliers) to deliver what we wanted and it was only when it was coming to really the end and we said (to end-users) 'look you've got to go with what you've got', and then having to roll that out to users was just so difficult. Business Manager (9)

The last comment epitomises the inadequate interaction between the two groups of users. From a practical point of view, a more 'hands on' approach would have been beneficial, not just from a user expectation perspective but also as a method of managing user acceptance. This issue is discussed next.
7.2.3 User Expectation - User Acceptance

User expectations have manifested themselves in terms of reduced levels of acceptance of the system, the most common complaints gathered through interviews reflect the disappointment experienced, terms used were:

- the system doesn’t work
- it doesn’t perform
- it’s slow, it’s too slow
- it’s unworkable
- this thing doesn’t work
- it’s taking 10 minutes to respond

Even though performance issues were addressed and system performance was dramatically improved technically, the mindsets of the general user community did not move on. Their expectations, generally, continued to remain higher than the actuality of the system. Analysis posits that the negative views held were formed at a time in the project when both internal and external factors were impacting on development damaging its creditability. For example the Client Project Manager commented:

_The problem is that it has been damaged by the fact that it didn’t work very well initially and that a lot of these people didn’t understand that any implementation like this has a bumpy ride and what they expected was perfection from day one._ Client Project Manager

The importance of engaging key stakeholders i.e. the user community, during periods of change cannot be denied, stakeholders need to be involved in all stages of a change programme if it is to be successful. For user acceptance to be successful valid knowledge is essential in order to manage the uncertainty that people feel. Analysis of data collected confirms that it was feelings of uncertainty that fuelled the negative views in the early stages of the project. Learning is a key issue in the need to build new mental models of the changing business processes and one that involves cultural issues. Without the learning there was insufficient
knowledge to engender acceptance and a sense of loyalty for the New IT System. Evidence clearly demonstrates that during the first part of the project loyalty to some extent remained with the Legacy System, and confidence of the new system was low.

The Client Project Manager reports that one of the projects within the Programme of Change was a 'Change Initiative' project. However evidence shows that the role of Change Manager did not continue throughout the life of the Programme of Change Project, and puts forward two reasons: (1) the initial Change Manager left the role during the initial stages of the New IT System project, (2) the in-coming manager did not adopt the same role due to the Foot and Mouth crisis at that time. Consequently when the project approached implementation there was no one in that role and this resulted in an on-going failure to effectively manage the business change aspects of the project, which would have informed user acceptance issues. Analysis posits that this is a key lesson to be learned. The researcher believes that the Change Manager role was needed to negotiate with the business users about critical changes whereas, in reality, it was the Client Project Manager who took on this role. Comments reflecting this are set out below:

There was a change manager in the Programme of Change for a while but then it stopped. CAPM Business (16)

... the IT project kind of subsumed some of the business analysis that should have taken place elsewhere actually. Client Project Manager

... what ended up happening was that the IT project succeeded in its main goals and was then used to drive some of the elements of the business change. Supplier Developer (8)

The above issues are associated with the nature of culture31, individual and corporate in particular that affected people's attitudes and behaviour operating

31 Individual, group and corporate culture together with bureaucratic are explored in Chapter 5.
within a bureaucratic environment. The researcher believes that, as with other aspects in this case study, communications provides a multi-link across the development factors within the research themes.

7.2.4 Training Issues

A fundamental facet of user expectation issues is that of user training. Although a sub-section of user expectation, user training is an essential factor that is critical to project acceptance and success. This view is put forward firstly by Martin (1991), and echoed by Markus (2004). Analysis shows that the delivery of structured organized training was problematical.

Within the context of the New IT System project user training reflected two levels, firstly the IT training that was going to be delivered in terms of system training and secondly, what business managers were going to deliver in terms of teaching people how to process a particular scheme on the new system. The new Generic Process involved new business processes that necessitated users moving away from their previous working habits and mindsets towards the new integrated working patterns of the New IT System. There is a direct link here to cultural issues in terms of the former organizational bureaucratic culture that was inherent in the project environment, and the group culture of the Business Manager imported into the development arena. This concurs with Hofstede’s (2003) view discussed earlier.

End-users expected to receive formal training in the new screen activities within a test environment before going live. This expectation was not realised. Even though training was accepted as an important factor and planned for, it proved difficult to execute because of the delivery delays experienced. Training was planned both
centrally in the project environment and on site in the DOs. Local Implementation Teams (LITs) were created where individuals attended centrally for training and then travelled around the DOs to cascade that training and knowledge to the end-users. This activity was unproductive; the perception is that there was very limited effective training that occurred in the DOs. Training could only effectively be rolled out in line with the incremental delivery of the system being developed. As slippage occurred, training was deferred. Consequently training issues fell behind. The occasions when training was being delivered on schedule were problematic because the new functionality and/or screen activities were not available to the users due to development delays – thus users experienced a further negative impact, they were not able to apply their new knowledge, for example:

... when people had been told 'you're going to be trained on the new system now, when you go back to your desk you'll be able to work on it' it wasn't there. Senior Manager (2)

The external delays, internal problems, time constraints and communication issues all impacted on the efficacy of the training delivered. Specific reference is made to the Foot and Mouth crisis that critically constrained the resources available to organize and coordinate training. The initial difficulties associated with user acceptance and usability of the New IT System are attributed to the poor levels of effective training. The Client Project Manager agreed that user training had not been as effective as planned. He comments:

... in terms of user training, I have a doubt as to whether we have delivered that effectively basically because we haven't put enough time and effort into that. We have had some very good trainers to be fair to them, I don't think they were nurtured enough or given enough information to do the job. Client Project Manager (1)

The above example is seen as a business problem because it is felt that more effort should have been made to get end-users involved. However other elements
that can be factored in reflect the critical time scales and late delivery. Deferral of training resulted in insufficient time to involve these users such that it was the middle users, indirect business users who made decisions about the acceptability of the front end of the system. Training was often ad hoc, unstructured and done at the last minute, one Business Manager commented:

*There seemed to be a lot of running around at the last minute, trying to organize training sessions.* Business Manager (16)

Managers commented that it became obvious that users did not understand the system, some of those that had received training either had forgotten it or not achieved sufficient understanding as the system was not being used properly. The issue raised through analysis is that the end-users did not understand their new jobs. In some cases users were still performing the paper-based tasks that were automated on the system. Therefore it is suggested that there was an inadequate understanding of the aim of the system and of what it was able to do. Furthermore, where understanding was present there was only limited confidence in it such that users felt the need to check manually that the system was performing the required tasks automatically 'behind the scenes'. The researcher analyses that it was difficult for the user community to grasp a holistic understanding of a system that was being developed and delivered in incremental stages.

### 7.2.5 Prototyping

As discussed in the literature review prototyping is a recognised characteristic of a RAD-type development approach and is especially associated with JAD workshops that are associated with user expectation (Alavi 1984, McConnell 1996, DSDM 2001, Maner 2002, Stapleton 2002). However, contrary to views expressed in the
literature evidence reveals that within the context of the case study use of prototyping was very limited.

External factors, discussed previously, prevented the project from getting into the powerful development iterations that had been scheduled and this had a major impact upon the prototyping aspect. Analysis reveals that due to the prioritising of the technical resource that could have facilitated prototyping, and the availability problems of the necessary business expertise at that time, coupled with the constraints of tight deadlines and delivery schedules meant that prototyping suffered. Techniques such as Use Case Modelling and Activity Diagrams evolved from the workshops and the 'Whiteboard' sessions that effectively replaced the prototype stage, and these were employed to demonstrate the system processes. Lack of prototyping meant that it was the mechanisms of system testing and user acceptance testing that were routinely used to verify functionality, and identify whether user and business needs had been met. One Supplier's developer working on the system screens commented:

"... in terms of actually designing the panels (screens) that you actually see... there perhaps was a lack of prototyping, it was more a case of they (the business managers) say what they need, it gets put on there, goes into system test or UAT and then it was battled out from there."

Supplier Developer (11)

The absence of prototyping caused disappointment to the user community. There was an expectation by the business managers that prototyping would provide them with a vehicle for feedback and control. However as this did not occur the conflict that existed between the Suppliers and the business managers, as discussed earlier, continued to cause difficulties. There was still a general feeling amongst business managers that had they had more control they may have been able to
prevent some of the delays experienced. Comments reflecting this view are set out below:

... we'd expected as a Business to see an iterative approach to the development, what I mean by this is I'd have expected to deliver a description of the business requirements to the project and I would have expected the project to say 'OK we've read and understood what we think you want and this is what we are going to build for you, here's a prototype', ... if I'd had that opportunity earlier on provided by the project I might have been better able to steer them in the right direction. Business Manager (3)

... what they (Suppliers) said was 'we will develop something for you on the basis of what you tell us in these workshops, we'll give you prototypes, you go away, you trial it and you tell us if you want this changed, that changed or it doesn't meet the needs', that never happened ... Business Manager (9)

... and yet that was sold to us (business managers) up-front, there would be this prototyping, and we've got classic examples of where it would have been beneficial. Business Manager (9)

Evidence supports the analysis that early in the project lifecycle, before the relocation of the project, PowerPoint presentations were used as throwaway prototypes. This style of prototyping is inconsistent with an iterative style development that utilises incremental development where the prototype becomes part of the final system, and illustrates the view that the approach applied was not pure RAD. A Business Manager describes one example of prototyping:

*That XX screen we (the business managers) saw once, it was a PowerPoint thing, it wasn’t the proper running prototype, we saw that once and that was it. There were a number of things that we said, ‘well as long as ‘so and so’ is alright and that gets done’ ... some of which I don’t think have arrived yet.* Business Manager (1)

Evidence puts forward the common consensus across the project population that prototyping was insufficient to provide real benefit to the project. Once the planned iterations were abandoned and development became reliant on requirements
elicitation and testing there was very little evidence of prototyping. Comments to support this as set out beneath:

I can't say that I have been in any meeting where prototyping was ever considered! Business Manager (3)

I don't think that it has been as successful as it could have been, I don't think that there has been much use of prototyping here ... I thought that in a prototyping environment, they could say I don't want 'that' there, I want 'that' over that side of the screen, and I think that that was what people expected to be going on but it's not happening. CAPM Business (2)

I don't think the prototyping we were meant to have had really happened to the extent it should have done ... what we have missed is the prototype stage. CAPM Business (18)

Prototyping necessitates a high degree of user involvement in order to clarify the system does what people expect it to do. Additionally it engenders a sense of ownership of the system by the users thus facilitating its eventual acceptance. Consequently the lack of prototyping can therefore be associated with the problems described in user expectation and user acceptance. This analysis is further supported by arguments put forward by Clegg and Barker (1994) who suggest that increased user involvement engenders familiarity with the system thereby reducing levels of training necessary. The single example of a potentially effective prototype was the Model Office which is discussed next.

7.2.5.1 Prototyping – Model Office

On the more positive side a Model Office was set up at the end of the first year. It was the first and only integrated prototype of the New IT System. It demonstrated the Generic Process concept and it brought together all the different elements of intended system. One purpose was to involve the user community more fully. However as the reader will now be aware there are a significant number of factors that are impacting upon the project and the Model Office was no exception. Once
again there were particular issues with time resources. The business managers were occupied with their Business as Usual activities and the on-going commitment to development. It was equally problematic getting the end-users, located off site in the DOs, involved such that demonstrations organized within the Model Office environment were not always productive.

I think part of the issue, again I remember specific discussions, was in terms of getting people out of the Business to spend some time in this sort of environment. Supplier Developer (8)

I think people just came in, looked at stuff, were probably (a) slightly tired at having to travel, (b) slightly intimidated by the environment, (c) didn't really understand what their role was other than to look at something. Senior Developer (8)

It only represented an end-to-end thin slice of the system, not 'a hands on', get involved and try it out experience. Consequently the opinions they took back to the DOs were negative ones.

...we had created a Model Office type environment ... so that the DOs could actually see the thing for real fairly early on and for a whole series of reasons that didn't materialise. Senior Manager (1)

All it was really used for was demonstrating, if you put a form in this end, a letter comes out this end, but they know that so they don't need to see it. Supplier Developer (9)

Analysis reveals that the utilization of the Model Office environment was instrumental in the low levels of success and productivity achieved. Far from providing a user training and operational environment where participants could interact with developing features, it turned into applications development demonstration environment. Initially the Model Office demonstrated the Generic Process to the user population who although able to see the critical business functions being performed were unable to take on board the conceptual elements behind it that affected its effectiveness. Set out below are comments gathered
through interviews across the project population, that express the general disillusionment attached to the Model Office environment.

*The Model Office as we saw it ... should have been a mini operational environment. It turned into a demonstration environment and not a very effective one at that.* Supplier Developer (8)

*The Model Office was quite a good idea, but it wasn’t utilised ...* Business Manager (14)

*I don’t think the model office was used to the extent it should have been ... It would have been very useful if the users could have seen screens in a prototype environment where they could have tweaked with it fairly easily.* Business Manager (18)

The business managers perceived the Model Office as a lot of ‘out of the box’ technology with a low level of customization i.e. it represented hi-tech components, not workable screens with actual data to demonstrate proposed performance. The transparency was insufficient to project the concepts supporting it, and in their view the model office was not put to best use. They refer particularly to the potential of prototyping interface screens and pieces of functionality, where users could have benefited from ‘hands on’ interaction. It was only in the UAT testing stages that they were able to see and understand the screen usability, how the data captured from the application forms appeared on the screen and the proposed operational functionality of the users.

Analysis shows that the Model Office environment was not effectively utilised as an integral and complementary part of the iterative development approach. It represented a lost opportunity and was subsequently closed down and the room was reassigned firstly to a meeting room, and later became a storeroom. It was the critically limited resources in terms of time and space that were put forward as a rationale for these actions. These events themselves demonstrate the level of commitment to, and importance of the prototyping concept within the project arena.
7.2.6 User Expectation – Conclusions

Prototyping is a recognised characteristic of a RAD-type development approach and is especially associated with JAD workshops that are associated with user expectation. Although the literature expresses the view that expectation management is a major success factor in a RAD-type development approach, evidence reveals that the use of prototyping as a mechanism to manage user expectation levels was very limited.

The failure to present a realistic account of the actual system being developed to the user community fuelled the high levels of expectation experienced. Project constraints and delays resulted in focusing on the delivery of the core elements of the New IT System, and this meant that the anticipated goals were not achieved, which when coupled with poor communications contributed to unrealistic user expectation levels that were left unchallenged. Analysis suggests better communications would have enabled a more realistic level of expectation to be cascaded to the others throughout the project, and thus the high-levels of expectation could have been more effectively managed. The fundamental failure to manage and utilise the Model Office environment as an integral and complementary part of the iterative development approach contributed considerably to levels of user expectation. Therefore prototyping was insufficient to provide any real benefit to the project.

This lack of integrated prototyping, and access of a ‘hands on’ approach to learning, together with reduced levels of training served to lower user confidence in the New IT System. RAD-type projects that are not accompanied by adequate staff training develop problems. Without proper orientation and involvement users are
less likely to understand the new business processes and new data activities (Martin 1991). Additionally pressure experienced through development delays and missed release schedules resulted in ad hoc and rushed periods of training that was further compounded by ineffective use of the Model Office. In analysis the need for a training environment is clear, particularly in light of the changes expected in business processes and working patterns that were crucial to the success of the system, and the successful acceptance of the system by the users. Training became dependent on the incremental delivery schedule, slippage occurred; training was deferred, fell behind and was perceived as inadequate. All of these issues impacted on user involvement, user expectation and ultimately on user acceptance of the New IT System. Evidence posits that neither the Suppliers nor the Department took sufficient responsibility in promoting actual user training either centrally or remotely. All of these issues impacted on user involvement, user expectation and ultimately on user acceptance of the New IT System.

7.3 Communications

This section analyses the communication factors present in the development environment and their impact upon the development approach. Analysis shows that communications were ineffective and had a considerable effect on the efficacy of the project environment. The analysis looks at how inherent civil service issues associated with such bureaucratic cultures have impacted on the communication structures both internally and externally. Discussion extends to the issues relating to the visibility and transparency of communications, aspects of shared experiences and knowledge gaps, together with problems associated with tacit, explicit and implicit knowledge from the project environment perspective.
7.3.1 Communications – Structure

Within the context of the research case study the Supplier's culture and the Client's bureaucratic culture stem from different domains, each of which was characterised by different behavioural processes, thereby identifying a particular need for communications. The Suppliers were working towards a clear common goal of developing a system, whereas the clients were working toward individual goals that seemed less clear, and which included a range of other business goals. The Supplier's prime interest was in the technology, whereas the Client was interested in getting a product that met their needs, which in the context of this case study were individual perceptions of what was wanted. Consequently communications were important in order to bring the different cultures together.

The Suppliers and the Department were co-located together within the centralised development arena, therefore it was the inherent organizational elements that had an important bearing on the design and structure of the communications. Within large and complex environments the degree of communications needed to be optimized. Coughlan and Macredie (2002) acknowledges that co-operative communications is not an option but a prerequisite for project success.

Communication is a multi-faceted topic area. The level of communication explored here relates specifically to the effectiveness of communications during the project across both internal and external environments that influenced the project development, and the people involved. Effective communication is a major component of a successful project. It is critical to user understanding à propos, changes in working practices, accepting new business processes, and promoting co-operation within the development project. Productive communications provides organization wide support, raises awareness and understanding that engenders
commitment from employees. As mentioned above the user community believe that ineffective and poorly managed communications were a factor that contributed to their high-levels of expectation.

7.3.2 Communications – Poor and Ineffective

A criticism that is acknowledged by the Project Board, and further recognised throughout the project environment, is that communications were ineffective and therefore problematic. However, it should not be said categorically that communications were ineffective and poor without first establishing the causal factors concerned. In analysis evidence supports the common vision that communications were unproductive. Informal communication channels that were identified epitomised the sub-cultures created by the project teams. This analysis concurs with the understanding put forward by Cockburn (2002) that mini-cultures sit within the culture formed within the larger organization and also within the dominant wider culture around it, teams will form their own patterns of communication.

Within the research case study project, formal communication structures were represented by vertical and horizontal routes across the project environment, and to the outlying CAPM Divisional Offices. Analysis of case material reveals that despite a continued drive to communicate by project management to the workforce, effective communications were not achieved. Evidence supports the presence of regular reporting though an intranet facility, progress reports, highlight reports, newsletters, information emails, workshop presentations, and some face-to-face disseminations throughout the project.
Additionally the co-location of the Suppliers and the Clients on the same site for the duration of the project facilitated informal communications between the business managers and the development teams, between the development teams themselves, and between individual business managers. Evidence suggests, and observations confirm that it was often quicker and simpler to directly approach people rather than go through the formal channels. This activity was particularly visible between the business teams writing the business rules and the Supplier's developers providing the underpinning coding. Moreover, informal communications also existed between the DOs themselves, and between the business managers and the DOs, these mechanisms were not created by the nature of the project but are representative of organizational culture and former practice.

To maintain a balanced perspective and present a coherent analysis of communication issues it is necessary to explore both internal and external communications. This section firstly considers the internal communications of the centralised project development arena, and secondly external communications representing channels between the project environment and the outlying organizational people. Some of the early attitudes concerning the perceived problematic nature of communications are documented through interviewee contributions.

7.3.3 Internal Communications

When analysing communications it is importance to realise that the message being communicated may not always be the same message that the recipient is receiving, this includes both written and spoken communication. It is clear that despite the communication mechanisms that were in place there was little effective engagement between the message and the audience. Within the context of this
research case study despite the presence of top down communications there was little evidence of either the receipt of those communications or acknowledgement of their content. These findings support the views expressed by Balogun and Hailey (2004), and Carnall (2003) that both the message content and its’ medium are key elements of effective communications. The perception was that communications were essentially one way – downwards. A key problem for the Suppliers was a fundamental lack of initial information sharing and feedback from the business managers during the early stages of the project. In reply business managers commented that contact with the Suppliers ‘just wasn’t there’.

Associated with this is the belief held by some business managers that communication between the development teams was poor. They maintain that it was apparent in meetings that their prior input to a particular development team had not been passed on to other relevant development teams. This was at a time when meetings were not integrated across the schemes but dealt with on an individual scheme basis. It was assumed that common data would be shared - this was not the case, for example:

... so you’d have different meetings with all of them and then you’d say ‘we told ‘so and so’ (Supplier’s Developer) that the other day’ and they’d say ‘oh he hasn’t told us that’.... Business Manager (3)

This is a very interesting point as it implies that the working nature of the Supplier Development teams epitomised that of a ‘silo’ style generally characteristic of bureaucratic cultures[^32]. However observations suggest that this relates to the formal communication methods, as evidence supports considerable informal communication between the parties working within the project arena. For

[^32]: Due to the constraints of the research project the researcher has not been able to explore this in greater depth at this time.
I'm not quite sure who was responsible for communicating what ... I always got the impression that they were one-way communication exercises, I didn't hear much coming back to the IT stream. Supplier Developer (6)

... it was picked up by us (Suppliers) that the communication process from Programme of Change to the Business was quite poor. (This interviewee confirms the same view in a subsequent interview) ... the external communication wasn't as good as it could have been but definitely, absolutely the internal communication was terrible ... Supplier Developer (8)

No, initially there was no contact at all with developers or the business analysts or very little. Business Manager (14)

As a consequence communication lines were regarded as vertical rather than horizontal and this did not facilitate easy diffusion of critical business data. This interpretation is put forward as an initial reason behind the limited understanding by the Suppliers of the business needs.

7.3.4 External Communications

The researcher's remit was to the centralised development environment therefore it must be recognised that the data gathered regarding the external communications is biased from an internal viewpoint. Evidence was collected from across the project community from people who had contact with and/or received feedback from personnel working in the DOs. However there was occasion for the researcher to interview and discuss with DO employees who visited the site.

A general consensus is that external communications between the centralised project environment and the geographically dispersed DOs were the least effective. There is a link here to the issue discussed above, that where the key
communication requirements were lacking then communication is unsuccessful. The people in the DOs felt they were not kept fully informed about project progress in spite of the mechanisms in place. Additionally the end-user community believe that their distance from the centralised environment resulted in a lack of support. They felt that they had no effective formal channels ‘backwards’, that they had no ‘voice’ such that their comments and opinions when expressed were unproductive.

It was the Supplier’s developers who expressed the most concern:

... a few times I’ve been out in DOs and a few times I’ve talked with DOs ... It all seems to be one-way, whether or not it is I don’t know, I get the impression is that it is. I don’t think they’ve got a real route to kind of raise the issues either. Supplier Developer (3)

I’ve only been out to DO a couple of times ... but I don’t think that those people have formal routes into the project .... so by the time their views get to the project they’ve been filtered and diluted quite a lot, maybe they need a more direct voice. Supplier Developer (9)

I think communication needs to improve ... you (DO end-users) will see this new function which will allow you to do that, ... I don’t see evidence of that ... I think they need that. It’s one thing I think is missing, communication out to DOs to say this is what’s coming or is happening, giving them some prior notice if you can. Suppliers Project Manager (2)

When analysing external communications it is important to consider the horizontal communications between the DOs themselves. Evidence reveals that there was little proactive communications between the DOs. This lack of interaction is indicative of their perception of not being a part of the project. The views and comments expressed below that illustrate this analysis are presented through the views and opinions of the business managers dealing with the personnel in the DOs:

...where, say one DO solved a problem but doesn’t pass the solution to other DOs. That is still poor. They have ... meetings to share best practice but I don’t think it works as well as it could do. Business Manager (3)
... the DOs felt isolated from the action, they saw the project as being centrally located, the people working there didn't feel involved and didn't have much commitment. Business Manager (16)

Poor communications can be illustrated through the following example. When the release of a piece of development work going into live production necessitated some related DO activity, business managers held presentations for the DO managers to inform them what would be required. They were also informed that they should expect a substantial response from their customers (the public) when it became public knowledge. The aim was for that 'new knowledge' to be cascaded throughout the DOs and their front office staff who deal with the general public. It was not successful, the knowledge was insufficiently disseminated and the staff in the front offices, who were unaware of the new requirements, were overwhelmed by the public response they were not expecting. A Business Manager who held those presentations commented:

Well basically we went round on XXX Scheme 2003 presentations to DOs ... said what to expect next ... need to get your front offices up to speed because you're going to have 100s of phone calls, basically where the communications fell down was that nobody told the front offices ... they had a 100 phone calls related to IACS questions which they were not expecting. Business Manager (16)

This example illustrates the breakdown in the communications chains that required remedial action to be taken and mechanisms put in place to prevent it reoccurring. It is also a further example of project management being more reactive than proactive. Analysis concurs with the findings of Cross (1998), who cites poor cooperation between developer and clients due to ineffective communications as main contributors to project risk.
7.3.5 Communications – Visibility and Transparency

Analysis suggests that an understanding of the project and its progress by the project community is associated with the visibility of communications – what is not seen or accepted as being communicated cannot be understood. The researcher therefore proposes the issue of visibility of communications as a point of discussion. Problems with visibility can be extended to include transparency - a relation of visibility. Where senior management believed there was visibility, the problem was more a question of transparency.

When applied to the ineffective communications experienced it can be analysed that although there was a strategy of promoting information that was designed to inform the project community, it was inadequate to convey sufficient understanding. Information itself is not necessarily transparent, it can confuse, deceive and mislead as easily as it clarifies. Dissemination of information was often not sufficient to enable understanding by the recipients. An illustration of this refers to the change in how the CAP forms were processed. Historically end-users would be responsible for processing a customer application through to payment or non – payment, but this was no longer necessary as the New IT System automatically processes claims through the stages of the Generic Process. Their responsibility is more related to problem identification and resolution. There is a view that the end-users did not fully understand their new roles. The following comment was made by a senior developer on the Supplier team but was echoed by other fellow developers.

*Users in the past have taken an application and processed it all the way through now they are not doing that and I don’t think they understand their jobs.* Supplier Developer (21)
Although there is some link here to training issues here, as previously discussed in Section 7.2.4, p245, the information disseminated during project development should have provided sufficient understanding of changing business processes and related activities.

7.3.6 Communications – Shared Experiences

It is accepted that face-to-face communications engender a better level of understanding about the problems and potential solutions involved in a development project, so that the sender and the receiver can develop a common point of reference (Cockburn 2002). This is representative of the case study and analysis posits that it is important to engage with this premise. Although there is some evidence of the horizontal communications within the centralised project environment being extended to the vertical communication lines of the DOs and the end-users it was not sufficient to achieve the purpose. Thus communications to this audience were largely textual. As a consequence any frames of reference and comprehension developed were not extended to the external environment such that they were not kept informed and therefore did not share the same level of understanding. They developed different interpretations of the ‘reality’ of the development project. Thus their interpretation, and hence their expectations, of the New IT System were different. This analysis is consistent with the findings expressed by Cockburn (Ibid.) and Balogun and Hailey (2004). This situation created a knowledge gap that fuelled the unrealistic levels of expectation experiences, and is discussed in the following section.

7.3.7 Communications – Communications Gap

Cockburn advises that ‘Communication is never perfect and complete’ (Ibid. p12). However people with similar or shared experiences are able to jump across an
information gap, the bigger the gap the more common knowledge is required. Therefore those without the common experience can only manage a small gap and require more information to achieve understanding. The argument is that although documentation is the least effective format of communications, in certain situations it can be the most appropriate. The need for face-to-face communications as a core mechanism is not argued here but where there is evidence of conflict and unrest, a less expressive communications channel, such as documentation, can be effective by removing redundant emotion. This provides the sender with an opportunity to reflect upon the message before delivery (Cockburn 2002). Within the case study there were instances in meetings where emotions ran high, people who felt threatened were reactive to stimuli that maximised the different levels of understanding achieved. Observational evidence reveals that this resulted in meetings becoming protracted and unproductive, consequently development decisions were deferred.

There are two points to be drawn from this discussion, the need to focus on an interactive exchange of information between people, and the selection of appropriate mechanisms to achieve this. Although Cockburn makes many references to Extreme Programming methods the researcher posits that the above principles are equally applicable to the development approach of the case study project.

7.3.8 Communications – Culpability and Justification

The expression ‘Communications is always one of those things that will always come up as an area you can always improve’ was a frequent comment that interviewees used to describe the communications situation. It serves to illustrate the recognition, and perhaps tolerance of inadequate past performance, and
acknowledgement that it needed improving. This comment almost became a standing quote from the project team to brush aside the identified issues lying within, for example:

You can always improve on communications and I think it's the sheer volume but we do as much as we can, as best as we can but you can always improve on it ... I think that's one of the, not the poor areas, but one of the difficult areas. Business Manager (8)

Contrarily to present a balanced argument, it is necessary to place a degree of culpability upon the end-users themselves, who Senior Management felt, did not take an active role in the communication processes, such that it only represented a downwards, one-way communications stream. The end-users did not challenge what they perceived to be poor and inadequate communications. Thus some degree of responsibility is theirs.

Management feeling is strong, even though it was an obvious failing of the project, they feel able to justify their actions as being sufficient for their perception of the needs of the project, and is demonstrated through the following comments:

People can make a lot of criticism about the communications but at the end of the day we were communicating, now if people chose to ignore it or if it wasn't giving them the information they needed well they should have ... there's a 2 way thing here, if it wasn't giving them the information they needed they should have been big enough to say 'well look this is great but could you tell us more about this or why can't you explain more about this to us'. Senior Manager (1)

We ought to have had a better communications strategy, I think we sort of had one but it was based on whatever was being done at the time. Senior Manager (2)

There is an element of defensive reaction attached to this issue. Nevertheless there was the awareness that poor communications culminated in indifference from staff who were not communicating, not listening, nor responding, then
adequate measures needed to be implemented to improve that scenario. This was not achieved.

7.3.9. Communications – Tacit, Explicit and Implicit Knowledge

Tacit knowledge represents the ability to perform tasks without necessarily being able to articulate how they are done (Polanyi 1967, Beynon-Davies 2002). Polanyi states ‘we can know more than we can tell’ (1967, p4). In other words it is knowledge that is not susceptible to verbal explanation in the same way that recognised knowledge is. Additionally tacit knowledge is rooted deeply in action tied to the particular activities of a specific context i.e. individual know-how (Nonaka 1991). For the Supplier’s developers this was problematical, as a significant part of the business process knowledge had evolved over time within the cognitive consciousness of key knowledge holders as tacit knowledge.

There was a need to convert it to explicit knowledge, a need to express the inexpressible. Nonaka regards tacit knowledge as highly personal (1991). In the context of the case study tacit knowledge represented the scheme specific peculiarities, their complexities and interdependent relationships. It existed as knowledge and experience in the individual business managers that has evolved over time in relation to the separate CAP schemes. In accordance with the views of Beynon-Davies (2002) the substantial tacit knowledge involved in the organizational work meant that requirements elicitation became problematical. In comparison the explicit knowledge of the business managers was more easily gathered because it was readily available, similarly their implicit knowledge was accessible through communication techniques such as questions, workshops and discussions.
To analyse this issue in more depth, it is necessary to go one stage further than looking at communications for communications sake. In some cases where lines of communication were evident, it is the quality of communications that is in question. By this the researcher means it is not sufficient just to communicate a problem or issue to someone, but for some forms of communication there has to be a resolution or response, some action or result attached to that communiqué. Otherwise it has no value; it is communications just for communications sake, for example a Supplier Developer commented:

_If I communicate an issue to someone then that needs to be resolved so that's where the breakdown happens. The line itself is OK, I know who to talk to, I'm aware of who's on the project and who can deal with which issues but, to me, for the line to be effective the issues need resolving._ Supplier Developer (20)

However, the researcher believes that the failure in communications is representative of their quality. Where communications carrying important messages were cascaded across the project environment, they were not of sufficient richness, or possess sufficient clarity of communication cues to enable clear comprehension. They did not stimulate an exchange of communications and or effective two-way interactions. This analysis is confirmatory of views expressed by Balogun and Hailey (2004).

This analysis is further extended to suggest that the channels used for the dissemination of information were not sufficient to engender an understanding aimed at reducing uncertainty. There is no evidence that it provided employees with the understanding they needed to fulfil their roles during the period of change, it only presented the information. However knowledge by itself does not enable knowing. The actual activity of interaction enables the transition of knowledge into
knowing i.e. ‘hands on’ use of system prototypes. Nonetheless, knowing should not be confused with tacit knowledge.

7.3.10 Communications – User Reception

An important factor that is linked into the limited reception of project communications is that of the level of pressure attached to the day-to-day activities. It was not possible to insulate users from their daily tasks. They did not feel they had sufficient time to take on board the messages that were being sent out. Delays and missed deadlines also had a part to play, if nothing was being delivered then there was no visibility to present, nothing was happening.

Additionally the culture of a hierarchical communication structure that was a feature of the bureaucratic development environment, proved challenging within the small teams that are characteristic of RAD-type development. It only produced limited feedback such that the users had an unrealistically high expectation of the systems’ functionality, and therefore had initial difficulty in accepting the evolving system. This analysis is confirmatory of research carried out by Jones and King (1998).

To resolve this problem a number of end-users were brought onsite to make use of a system prototype, the Model Office, towards the end of the first year. However as discussed in Section 7.2.5.1, p250, the experiences and knowledge of the developing system they disseminated back to their colleagues was unfavourable because their perception of the system was limited. They were not able to transform that knowledge into understanding and this affected their attitudes and acceptance of the system. There is a direct link here into the lack of initial user involvement.
7.3.11 Communications - Conclusions

It is accepted that communications were both poor and ineffective despite the efforts made by project management to disseminate and inform through a variety of mechanisms. The poor and ineffective communications influenced the project development across both internal and external environments from vertical and horizontal perspectives and this affected people's attitudes and morale. A high-level strategy and a realistic, practicable framework that provided a two-way communications channel for dissemination and feedback were found to be wanting. Within the context of this case study there was limited evidence of high-level structured control and no overall communications manager. This situation resulted in a lack of continuity and an ad hoc approach to communications.

Where lines of communication were evident, it is the quality of communications that is in question. In some cases it was communication for communications sake. Where communications carrying important messages were cascaded across the project environment, they were not of sufficient richness, or did not possess sufficient clarity of communication cues to enable clear comprehension. They did not stimulate an exchange of communications and or effective two-way interactions as suggested by Balogun and Hailey (2004).

The lack of effective communications essential for a successful project prevented users from understanding, and accepting changes in working practices and new business processes. It further inhibited the promotion of organizational wide support and the awareness, learning and reception of knowledge that could have engendered high levels of commitment from the employees affected. Ineffective and poor communications also restricted knowledge sharing and did not facilitate
easy diffusion of critical business data and consequently led to inaccurate requirements that impacted upon development schedules. The communication mechanisms put in place were not sufficient to facilitate effective engagement between the message and the audience. Issues of the visibility and transparency of communications meant that the dissemination of information was often not sufficient to enable understanding by the recipients, and this created an on-going communications gap that fuelled the unrealistic levels of expectation experienced, which could have been addressed through prototyping activities.

Even though prototyping is a fundamental characteristic of a RAD approach (Martin 1991, Crinnon 1992, McConnell 1996, Beynon-Davies 1999, DSDM 2001, Whitely 2004) there was very little evidence of its application. As the literature suggests what would have made a difference would have been more visibility of the product and more opportunity to see development progress through the use of prototypes that could have reduced the associated risks (Vonk 1990, Beynon-Davies 1998, Tudhope et al. 2001, Stapleton 2002). This lack of prototyping reduced the potential for communications in terms of meaningful feedback that is seen to contribute towards successful systems development (Alavi 1984, Coughlan and Macredie 2002).

However, it is recognised that the end-users did not challenge what they perceived to be poor and inadequate communications. Responding to pressures of work in relation to Business as Usual activities also contributed to low reception of communications because it was not possible to isolate people from their daily tasks such that users had sufficient time to participate actively in the communications processes. This echoes previous conclusions drawn that there is
a need for key users to be released from daily tasks and committed full time to the development project.

7.4 Chapter Conclusions

This chapter has discussed aspects of systems development that were pivotal to the system being developed in terms of user involvement and user expectation together with communication issues. Initial low levels of key knowledge holders and end-user involvement meant that development did not get into the powerful iterative cycles that had been planned. Restricted user input meant that the system scope was inaccurately defined and inaccuracies experienced during user testing served to reduce user confidence of, and commitment to the new system, and created unrealistic expectations. Both internal and external communications are accepted as being poor and ineffective, and had a considerable impact upon the project, which to some extent can be aligned to the bureaucratic nature of the organizational culture.

The issues analysed and discussed in this chapter are the subject of conference papers that have been presented and published in the official conference proceedings. A complete list is presented in Chapter 9, Section 9.5, p348.
8. Introduction

This chapter sets out the second part of the analysis of the case study material gathered. It continues with the analysis framework set out in Chapter 6. It turns its attention to the remaining three development issues of requirements elicitation, decision-making and testing that affected the research case study development project.

This grouping was also influenced by the natural divide of the data in the context of the research case study. Requirements elicitation was closely linked to decision-making and testing issues. More specifically it was the same organizational people engaged in defining the business needs that were involved in the decision making processes and additionally, linked to communications issues. Although communications have been analysed in the Chapter 7 analysis, it is recognised that issues surrounding the quality and effectiveness of communications also impacted on these areas, and therefore will be discussed in context, but not separately.

Analysis focuses on problems experienced with the requirements elicitation through examination of the JAD sessions and timeboxing concepts of a RAD-type approach and their influence upon development scope.
It further looks at how cultural constraints surrounding the decision-making processes of the former bureaucratic environment impacted upon the need for fast authoritative decisions, and the steps taken to resolve this issue that were unsuccessful. Also the difficulty experienced with an external stakeholder, and the 'fit for purpose' concept associated with RAD-type approach.

Lastly the issues relating to user acceptance testing, the creation of a joint testing environment and the difficulties associated with unrealistic testing schedules that led to delivery delays are explored.

8.1 Requirements Elicitation

A RAD-type approach requires that end-users participate strongly in the non-technical design of the system being developed, with particular reference to structured JAD workshops. The requirements specification for the Generic Process had been baseline at a very high-level that conforms to the 7th DSDM principle. However, from the business managers' points of view, this high-level view did not represent sufficient detail to meet actual business needs. Although it represented the common activities across the schemes it did not take into account the individual requirements of the separate schemes, nor the complex interdependent relationships that existed between them. Thus during the initial development stages of the project, integrated teams of Suppliers and business managers jointly participated in JAD workshops to facilitate understanding and requirements gathering.

Requirements elicitation processes were subject to immense tension within the project environment. As discussed in the Chapter 7 analysis, conflict was evident between the Supplier's developers and the business managers during the initial
requirements gathering stages. The conflict represented a divide in the understanding of the level of requirements that were deemed necessary. Literature advises that the degree of understanding depends on the way the requirements are communicated, the richness of the data gathered and, the sharing of that data (Coughlan and Macredie 2002). However within the context of the large and complex project environment these elements were difficult to achieve.

Involving users in the design and development processes is accepted as a more effective way of understanding user requirements, particularly during an iterative development approach. This understanding is in contrast to the one-to-one interviews/meetings of up-front requirements gathering favoured by the more traditional linear development approach. Within the case study scenario it was evident that JAD workshops were used for requirements elicitation during the initial stages of the project. Such sessions were used as the mechanism for specifying the crucial business knowledge of the individual scheme experts. However, as stated above, difficulties arose between the Suppliers and the business managers regarding requirements elicitation that impacted upon the project.

Evidence, both documentary and verbal, confirms that although the set up of the JAD workshops was typical of that expected with this genre of development i.e. integrated teams of Suppliers and organizational people, a facilitator, a session leader and so on, their size can be questioned. In the first instances the workshops were fairly large gatherings of 12+ people that were representative of all the business areas, rather than the small groups of 4 – 6 people more usually associated with JAD. As a consequence they were less productive. Issues surrounding JAD are discussed in the following Section.
8.1.1 Requirements Elicitation – JAD Sessions, timeboxing

As expected with a RAD-type approach, JAD workshops were combined with the characteristic of timeboxing that applied fixed deadlines as mechanisms for handling flexibility of requirements within such an iterative development environment. The theory is that system requirements are prioritised in line with the MoSCow\textsuperscript{33} rules concept of this genre of development approach.

In the context of the research case study, however the time boxing element, where the JAD workshops were run with set objectives tied to a specific timeframe, caused a lot of consternation for the participants. They found the concept of not being able to complete the set objectives inside a given time period, and letting go of the workshop activity without a subsequent planned activity to achieve the remaining objectives, difficult to operate within. A common view is that, as a consequence of this, compromises were made to meet timeboxed sessions by people who did not fully understand the implications of the compromises made. This view was compounded by the general opinion that the time scales themselves were both too aggressive and unrealistic, and thus pressurised participants’ decision-making. This state of affairs confirms opinions expressed in the literature, for example Martin states that ‘A timebox approach … does not work well if there is pressure to meet an impossible deadline’ (1991, p220).

Analysis shows significant critique is attached to the efficacy of these JAD sessions. When first questioned, in every instance, both the Suppliers and the organizational people referred to them as ‘difficult’. Evidence posits that workshops are not common practice within the hierarchical civil service culture and

\textsuperscript{33} MoSCoW rules, a component of RAD, are described in Chapter 3, Section 3.5.4.2, p95.
therefore in some cases people’s unfamiliarity with workshop processes prevented them from seeing the potential benefits of such sessions. Consequently there was some initial resistance to them. This is confirmatory of Martin’s (1991) view that for a workshop scenario to succeed it needs all participants to have positive attitudes.

The move from the previous siloed pattern of working towards the new integrated team working environment meant that people experienced uneasiness in the team workshops, they felt outside their comfort zone. Consequently decision-making was problematic, as people did not want to present their ‘thinking’ in front of their colleagues. Analysis of data gathered posits that they did not feel able to make the necessary decisions because culturally they were used to deferring such tasks up the management line. It is evident that they did not feel they were on an equal plane, and therefore felt unable to contribute effectively. The issues associated with effective decision-making are further explored in Section 8.2, p289.

An additional argument to be made reflects how difficulties surrounding business requirements elicitation impacted on development and delivery deadlines. With timeboxed scenarios when requirements were not forthcoming, regardless of whether a particular deadline was immovable or not, those deadlines coming up behind remained rigid. Therefore any delay on one deadline would consequently impact on those further down the line. In reality, although the timebox remains static the scheduled work does not - it just gets descoped. Thus the timebox itself is met but if there is too much descoping then the amount of work that is actually delivered can be negligible. Evidence reveals that this scenario was problematic for the Supplier’s developers, who commented:

One of the things that seemed to happen before, probably nearer the start, was trying to get that information by a certain deadline and if it
wasn’t there by that deadline what was the implication on all the other deadlines. Supplier Developer (11)

Consequently this situation is thought to have fuelled some of the development delays experienced, and is also connected to ineffective decision-making that is analysed in Section 8.2, p289.

An additional pressure materialized from the Political arena associated with the Department. The directive to meet critical EC deadlines was passed down through the Senior Management Team to the development arena as slippage started to occur. This meant that there was no opportunity for the project to take stock and reflect or, to progress development at a more realistic pace, thus increasing the pressure to deliver on schedule. One example is set out below:

*We came to XX scheme payments which had about 4-5 weeks to run, and we said ‘we are not going to make this’, ‘but the Minister had said that we’ve got to’ …* Suppliers Developer (21)

The above example resulted in raised levels of anxiety and stress. This particular piece of development work got forced through, and the deadline was achieved. However this result was at the cost of other development work that was deferred and scheduled release dates delayed.

In analysis it seems that a lack of proactive leadership towards the JAD workshops, and of the management of people’s activities within the JAD sessions would have facilitated more focused productivity towards development objectives. Additionally more engagement with Ministerial directives earlier in a schemes development timeframe may have permitted more realistic scheduling. These
issues are linked to decision-making difficulties that were experienced and are
explored in Section 8.2 and its sections.

8.1.2 Requirements Elicitation – Cultural Constraints

A second factor that impacted on requirements elicitation can be related to
organizational cultural issues. The case study project was operating within a
hierarchical environment and an associated vertical reporting structure. Where
employees felt they were answerable to their line manager there was a perceived
apprehension in speaking ‘your’ mind. This is similar to findings of Barrow and
Mayhew (2000) who suggest that where democratic structures are not in place
then ‘rational argumentation’ is difficult and ‘it is unlikely that less powerful
individuals will always be able to participate effectively...’ (p100). Several of those
interviewed on the organizational side were reluctant to voice opinions in
workshops if their managers were also in attendance. Business managers report
that they preferred to consider their ideas and their responses before making
decisions in front of their superiors, as was the custom of their former work
patterns. Although there is a relationship here to the issue of selecting the right
users, the rationale behind these issues was culturally based. For example, those
present in the workshops with the right knowledge did not always feel that they had
the authority to make a decision. In the same way some of the higher level
managers who had the authority did not feel they had sufficient business
knowledge to make the necessary decisions. These people were not necessarily in
the same meetings. The researcher puts forward the following comment from a
Business Manager who sums up all of the above points:

...the difficulty we had in the workshops was that the people who had
the day-to-day experience and detailed knowledge of what our
requirements were, were ... (names removed), but they didn’t have the
authority to say ‘do that’ and that is not necessarily a bad thing because
those people in fairness to them would have been speaking from their own personal experience of the schemes they have experience in and not across the Business. Business Manager (6)

When questioned the Client Project Manager commented:

... we were holding meetings and people would attend because they wanted to be involved but they actually wouldn't make a decision. So we wanted to create Business Champions\(^\text{34}\) to start making decisions basically. The reasons that's failed in most cases, but perhaps not all, was because those individuals didn't want to make decisions or couldn't. Client Project Manager

Such was the seriousness of this situation that a number of Supplier personnel also commented on this state of affairs:

There's definitely an attitude of not wanting to criticise your boss ... that would just be a comment that is not of the same opinion of your boss, that seems to be perceived as a criticism so there isn't that openness of being able to comment or speak their mind. Supplier Developer (6)

The culture in government and also the Department is the seniority in the room and having 20 people to a workshop where only one person speaks and they happen to be the most senior person in the room isn't helpful ... I think the culture of being almost subservient to your masters in the room it ... wasn't a culture that was helpful in the beginning, and then trying to get decisions was also difficult ... Supplier Developer (1)

The literature review examined the importance of end-user engagement in the requirements gathering process. As discussed previously the Foot and Mouth crisis impacted on the allocation of users and their availability to the project. This situation affected requirements elicitation in terms of the lack of specific end-users. Thus the Suppliers experienced difficulty in obtaining the requirements such that it was the business managers who initially provided the requirements, for example:

One of the unusual aspects of this has been that we don't really have any involvement with end-users as such. The requirements that we are given are given to us by process managers, occasionally we've set up workshops with process managers and the process manager may bring along somebody from DO but it is fairly rare. Supplier Developer (3)\(^\text{35}\)

---

\(^{34}\) Business Champions are further explored in Section 8.2.2, p292.

\(^{35}\) This quote has also been used on p219 to illustrate a related point.
I don't think it can be underestimated the amount of impact that that had. (i.e. the Foot and Mouth crisis) It just made running workshops difficult to impossible because you couldn't get the people because they were involved in more urgent work elsewhere Business Manager (11)

I know for a fact that when we did the initial 3 month workshops, the feedback I was getting was that it was very difficult getting the information. Suppliers Manager (1)

While the Suppliers accepted that more detail was needed, it was at a time when their development focus was trained on the Generic Process. As discussed previously analysts at that time, adhered to their pre-defined scope regardless of the increasing business manager's demands for greater detail to meet business needs. They were reluctant to collaborate with business managers outside these set parameters, they were working to a fixed schedule of activities. This caused a lot of angst and concern for the business managers that led to tension and conflict as already mentioned. As a consequence of this, there was no common vision and therefore no team identity, unity or spirit, there was no 'we' and ultimately no trust which fuelled the problems around issues of scope. This is illustrative of views put forward by McConnell (1996) and Balogun and Hailey (2004). Business managers felt very strongly about this issue and the following comments illustrate their frustration and resentment surrounding requirements elicitation and scope, for example:

... we would have the discussion with the Business Analyst (Suppliers) and it was as if there was a predefined end product, in that you'd have a discussion and it would be 'you can't have that', and 'you can't have this, this is what we think you need ... Business Manager (14)

XX came across as so self assertive and confident that in a lot of cases you thought he was right and I don't think XX was prepared to listen enough to business experts and now there are problems coming out because of it in certain areas. CAPM Business (2)

... I think it was the fact that what we were saying wasn't particularly taken on board and what were asking for either you can't do that or, there was a lot of 'that's out of scope'. Business Manager (9)
There are plausible links here to the issue of identifying the ‘right’ users for a project team, and to issues surrounding development scope discussed in the Chapter 7 analysis, Section 7.1.1, p218.

8.1.3 Requirements Elicitation – Scheme Complexity Issues

Some authors have associated the iterative development nature of a RAD-type development approach with a reduced need for specifying business requirements (Martin 1991, McConnell 1996, Linthicum 1997) through JAD workshops, as each development iteration amends the deficiencies of its predecessor. However in the context of the case study project where the planned iterations did not occur, the need to specify business requirements was critical to development. Although the Requirements Catalogue defined the high-level system requirements, it did not accommodate the degree of detail needed by the business managers, consequently a high rate of change requests occurred. Thus the under scoping of the project meant that the development schedules were overly optimistic. This undermined planning, and affected the upstream design and development activities. As a consequence it then necessitated rescheduling of both release schedules and resources, effectively contributing to missed deadlines, for example:

*Some of the planning has perhaps been optimistic to say the least. We keep seeing release schedules that are not met. Business Manager (7)*

Additionally, analysis of interviews with Supplier personnel reveals that in the early stages there was little understanding of the interdependent relationships that existed between schemes themselves or the related EC legislation. Although aware of a high-level Requirements Catalogue presented at procurement there
was no realization by the Supplier's developers of the depth of detail that was needed, for example:

... the examples (of schemes) I was led to believe we were given in the bid process didn't hint at 1% of what they do now ... Supplier Developer (6)

From the procurement process the Suppliers made assumptions about the system. They adopted a broad and over simplified view of the system. It was seen as representing a system that would accept a grant/subsidy application form, carry out validation, and process a payment in respect of that application. Their expectation was to generate an application form, capture the data back from each form, do some processing on the data captured, and then make a payment at the end.

However a major part of the complexities concerns the validation processes of the individual schemes, they are not all validated in the same way. Added to this was their particular related regulatory legislation that required complicated computational calculations. Consequently this lack of understanding, or exposure to that information early in the project development lifecycle was a causal factor in delayed release schedules and missed deadlines. A senior manager on the Suppliers side commenting on their developers confirms:

I remember them (the Supplier’s developers) saying, scratching their heads and saying that they didn’t know that these things inter-linked, no one had told them up to that point that they inter-linked, and they were scratching their heads and trying to find a technical solutions how can we make this work. Suppliers Manager (21)

It is possible here to link this point to the analysis reported earlier of the exclusion of the former IT people whose experience and knowledge of the business area would have been invaluable. Another interesting point to raise here is the lack of domain specific knowledge, i.e. an agricultural context, of the Suppliers meant that they were subsequently unable to direct requirements gathering if necessary.
Furthermore, assumptions were made that the scheme process definitions for the Generic Process would be provided in sufficient detail at the beginning of development by the business managers during the JAD sessions, and that would enable the Supplier’s developers to proceed through to the build cycle. In reality this proved not always to be the case and what should have been a relatively simple business process often turned out to have major complexities that impacted seriously on the development timeframe. A criticism offered is that insufficient effort was put into analysing the business requirements at the front end of the project such that the development schedules were unrealistic. This was compounded by the lack of awareness of the complexities involved.

There is some contention over the view of whether the complexities were in fact communicated to the Supplier’s developers. As mentioned above the developers affirm that they were unaware of the extent of the complexities involved. Contrarily the business managers report that they highlighted the inherent complexities but experienced initial difficulty in getting the Suppliers developers to understand the extent of those complexities. A Business Manager involved with one particular scheme development commented:

... we went to workshops and we certainly explained the complexities and how difficult it is and I think the personnel at the time said ‘no, no, you've just made it complex’. Business Manager (9)

Analysis reveals that at this time the development focus was on the Generic Process, not on the separate scheme requirements considered to be outside that remit. Additionally there was an expectation by the Supplier’s developers that any legislation and scheme data complexities would be dealt with through the application of business rules development. The researcher analyses that the Supplier’s developers were not given sufficient information about scheme detail
early enough in the project lifecycle to have been able to appreciate the full extent of the complications involved. It was not until requirements drilled down to the business rules level that they began to comprehend what was involved. Another Business Manager involved with scheme development commented:

I think that as a Business we were promised 'just give us your rules and we'll knock them through the system, and Bob's your uncle'. We knew the complexities of our schemes but we thought 'well clever people, clever technology lets give them the rules and see what they can do'.

Business Manager (6)

It is not until you get down to the level of business rules that you could see the faces of project members the penny's dropping as to how complex some of these schemes really are. The disbelief in their faces when they find out what the EC are actually obliging us to do...

Business Manager (6)

It is the business rules that contain the expert knowledge and accommodate the complexities involved within the CAP schemes. Thus the business logic is developed into the system that reflects relevant conditions and actions of the schemes as rules, which are then launched by the Business Rules Engine during run time.

However, case study material clearly demonstrates that when the Generic Process was being developed the scheme complexities that were crucial to that development were not taken on board even though the business managers believe they were communicating the complexity issues to the developers. The developers may rationalize that the complexities were under represented but at that time they were focussing on the generic processes across of the schemes not their separate peculiarities. Linked to this concern are other issues discussed within this research case study, such as the ability of the Business people to participate, co-operate and share knowledge, ineffective decision-making, poor communications all contributing to areas of conflict.
A related problem concerns the need to maintain the day-to-day activities of the current schemes. There was a critical need to achieve Business as Usual EC payment targets regardless of the state of scheme development. Therefore where there was a potential risk of missing crucial payment deadlines it was necessary to put contingency plans in place. Consequently when scheme development work approached the related payment deadline, the development focus was split between progressing scheme development work onto the Generic Process and, to preparing the Legacy System to take over that schemes payment tasks 'just in case'. This created a paradox whereby the Suppliers were working hard to produce the new system to replace the Legacy System, whilst also circumventing that work to ensure that the Legacy System could take over if delivery deadlines were not met.

A specific example of this refers to the development of one particular scheme when development onto the New IT System was delayed such that the risk of missing the EC payment window was realised, and it therefore had to be returned to the Legacy System for processing. There is an important connection here to not utilising the former IT personnel's experience and knowledge. It is suggested that they would have had some awareness of difficulties being experienced during system development.

From analysis a further point needs to be made in relation to requirements elicitation that is associated with the exclusion of the former IT expertise. Had the former IT staff with the inherent business knowledge and related IT experience been involved, it is speculated that for this piece of development the Supplier's
developers would have been more aware of the complex issues involved which had delayed development, and this situation may not have occurred. As discussed previously their experience and expertise would have been invaluable, for example:

... the knowledge more often than not was within IT because they would understand the capabilities of what changes could be incorporated, they would understand better more often than not than anyone in the operational branches ... I think that in all fairness as well that there is a certain under estimation from the suppliers of the task ahead. Business Manager (10)

In the old days 2 of the 3 teams (IT) we had were very experienced in CAP Schemes and all of agriculture really, they had worked on other computer programs for us, so they had a feel for what we want as well. They could also tip us off with potential problems, they could see what would cause a problem say with (EC) statistics for example. Business Manager (12)

Associated to this issue was the recognition by business managers that it was not until they attempted to express their own areas of scheme expertise did they realise that their IT related business knowledge was incomplete\(^{36}\). A percentage of their understanding was acceptance for the way things were, the way IT processes worked – that knowledge was in the former IT department. This is demonstrated through the following comments:

... you think you know your scheme but it’s only when you try and articulate them and define rules for those processes you actually... that there is a lot that you took for granted. Client Project Manager

When we came to realise our requirements for the new system we quickly realised that we didn’t know our schemes as well as we thought. The computer programmers knew them but as a Business we didn’t, now that’s one aspect. Business Manager (10)

Formerly the Legacy System had a direct business link between the Departments’ business analysts and the IT department. This is not the situation with the New IT System, it is the business managers who liaise with the Suppliers who then

\(^{36}\) This refers to tacit knowledge and is discussed in Chapter 7 Analysis, Section 7.3.9, p262.
develop the requirements. Consequently the former IT people in some cases had a better understanding of how the schemes interrelated than the business people.

8.1.5 Requirement Elicitation - Conclusions

Within the context of this research case study, the development factors of requirements elicitation, decision-making and testing are all related. They are influenced by cultural issues and linked through the use of timeboxing and descoping elements that are important factors for a RAD-type approach.

Early requirements elicitation was critical in the initial design and development stages of the project to ensure that the system developed was both ‘fit for purpose’ and met the dual objectives of user and business needs. However a more collaborative working approach was required such that both user contexts and organizational factors could have been realised and thus increased the degree of success achieved.

Significant critique is attached to the efficacy of the JAD session due to their unfamiliarity within the hierarchical civil service culture that in some cases prevented their potential benefits from being achieved. The JAD workshop activities proved problematic in terms of participation and productivity that was realised through the failure to identify the levels of complexity, and the relationships that existed between the schemes. The failure of the Supplier's developers to initially acknowledge the level of complexity involved, together with the unrealistic timeboxed release schedules led to problems with development scope that fuelled the existing conflict between the business managers and the Supplier's developers.
It is argued that a more traditionally structured and formal requirements analysis in the early data gathering stage, before applying iterative development cycles, could have revealed the inherent complexities that existed between the schemes allowing more realistic development schedules to be applied. Thus the under scoping of the project meant that the development schedules were overly optimistic. This undermined planning and affected the upstream design and development activities and necessitated rescheduling of both release schedules and resources that effectively contributed to missed deadlines and impacted upon the effectiveness of the iterative development process. With regard to large and complex systems, the more complete the requirements the more successful the system is in terms of user expectation and user acceptance (Coughlan and Macredie 2002).

It could also be argued that the outsourced Suppliers were chosen as experts in their field, and thus it was their responsibility to investigate and verify for themselves what was required for the system, what levels of requirement gathering were necessary, particularly in light of the controversy that existed within the workshops scenarios. Upon recognition of the complexities involved and the impact that would have on project scope in terms of design, build, release, and system testing it may have been prudent for the Suppliers at that point to have stopped development and renegotiated the schedules. An interesting inference is that had the former excluded IT staff with the inherent business knowledge and related IT experience been involved then the Supplier's developers would have been more aware of the complex issues involved.

Requirements elicitation suffered from cultural difficulties in terms of the unfamiliarity of JAD sessions, participants struggling to voice opinions in front of
their superiors and the availability of the 'right' users were all challenging. Where employees felt answerable to their line manager there was a perceived apprehension in speaking 'your' mind. The consequence was a lack of common vision and limited team cohesion that hindered trust developing in the crucial early stages of design and development.

In conclusion it seems that a lack of people management and of proactive project leadership enabled the Business people to adhere to their former working patterns and individual prioritises that led to decision-making difficulties that directly affected the ability to achieve the iterative and incremental development cycles.

8.2 Decision-making

The RAD-type approach relies heavily on the ability of those concerned to make empowered decisions in a timely fashion, without having to resort to higher management for guidance or control. This was very problematic within the project arena that was historically bureaucratic by nature. Business managers were authorized to make decisions but often found it difficult to make the transition from the previous behaviour of referring decision-making up the line management hierarchy, to the fast authoritative decision-making required.

8.2.1 Impact of Bureaucratic Culture on Decision-Making

A key problem experienced that had a noticeable impact on development, and which can be measured in terms of missed delivery deadlines, relates directly to former bureaucratic behaviour i.e. decision-making. Teams that are empowered to make authoritative and timely decisions without frequent recourse to higher management is the second of the DSDM principles. Within the case study project even though key people were authorized to make fast authoritative decisions it
proved to be a major problem. Historically, the organizational structure was typical of most Government departments and organized into a hierarchical line management structure where people reported directly to line managers working within a perceived ‘blame culture’ environment. Thus decision-making tended to be deferred up the management line. It was noted that some members of staff who expressed anxiety about the proposed business changes had difficulty ‘buying into’ the Generic Process concept and saw it as counter cultural, comparing it to previous work patterns. Consequently, even though the project consisted of integrated development teams there is a perception that these managers were still working with mindsets of the former scheme specific management that was necessary for their continued day-to-day activities on the Legacy System rather than the Generic Process.

In the past the CAP Schemes were dealt with on a scheme basis and were the responsibility of individual Process Managers and Scheme Managers. They attended to the scheme business needs and administration and were deemed to have ownership of the business processes involved. These people were identified as the key knowledge holders and decision makers who held the necessary understanding and detailed business knowledge that was required for the Generic Process of the new system. They are seen as the indirect users of the system compared to the direct end-users who perform the daily scheme activities.

Observations confirm that this problem was particularly visible in development meetings where difficulty arose in the prioritising and subsequent scheduling of scheme development work. In scheduled meetings those business managers present believed their own priorities to be paramount and although required to make decisions, did not feel able to do so if they felt it was counter to their own
individual agendas. The absence of senior managers in meetings meant that there was no one present with sufficient authority to impose decision-making and direct development activities. This was observed by the researcher and confirmed through interviews and conversations with both Suppliers and business managers, for example

... actually what happened was everybody was still saying 'my priority is first, mine's the first', from 5-6 different Business Leaders. Suppliers Manager (21)

However an influencing factor reveals that the business managers are deemed to 'own' their scheme business processes and as such did not respond to the level of supervision that was applied. Consequently, people management is viewed as reactive rather than proactive and considered weak since the authority exerted was not sufficient or effective enough to handle the impact this issue had on initial development delay.

A further difficulty related to this issue was getting agreement from the managers about what was core to development and what was secondary. The inability to make empowered decisions about business needs was a key concern for the Supplier's developers who needed prioritization of development work to meet timeboxed development deadlines. There is evidence that for some managers cosmetic changes to the system were as important as getting a fundamental aspect of the system working. Comments from the Supplier's development team echo the example set out below:

I think it's very difficult to keep the Business on track during meetings, they do tend to wander off and try to solve every single little problem. Supplier Developer (17)
Another problematic area that affected the decision-making process concerned the ability of business managers to attend the regular development meetings due to Business as Usual work pressures, and the logistics attached to those distantly located in the DOs, for example:

... also you find that you do sometimes go into meetings and there are a lot of project people, a lot of development people and no business people or very few. Supplier Developer (17)

... my branch is spread across the whole region ... obviously even a meeting is a total day out for everybody because of the distances everybody has to travel. Business Manager (B)37

8.2.2 Decision-making – Business Champions

The impact of ineffective decision-making was severe enough for the roles of Business Champions to be introduced as a potential solution. The intention was for key business users to be appointed responsible for a key business or functional area such that they would become empowered to make pragmatic decisions on behalf of the CAPM Business in order to move development forward. However, this concept was also a failure. The reluctance to take decisions continued. Supplier’s developers commented:

... we tried to have particular sorts of people in the role of Business Champions to be representative of not simply a Manager but somebody who was in a particular role, told they can take decisions ... that just didn’t work. Supplier Developer (8)

... they were empowered to make decisions but they just couldn’t. It was very frustrating we were trying to meet deadlines but this, well it just make it very difficult. Supplier Developer (21)

Analysis determines that the aim behind the creation of the Business Champions was to move away from the previous situation of being authorized to take decisions, to one of empowering people to make decisions. A further objective was

37 A different identifier has been attached to this comment to prevent recognition to other quotes with the same reference number, because it is identifiable to a specific person.
to engender motivation, and encourage the move away from a risk averse culture to a more risk taking environment. However, there is a fine distinction between being authorized and empowered to do something. Within the Department authorization carried an implied responsibility whereas, in the context of this case study, empowerment reflected the ability to do something because it was necessary to progress development. The former adhering to the idea of blame, the latter designed to remove the blame issue and emphasize a 'learning from mistakes' culture. An important objective was the implied removal of the blame perception from the empowerment status that accompanied the authorization status. This was the interpretation that was understood by senior management. However, the researcher believes that the subtle nuances behind this problem were not effectively communicated to those involved. There was evidence of a continued adherence to the blame culture concept, for example a Business Manager commented:

I was a Business Champion ...there was no iteration we were just puppets really, we were Champions so they could blame somebody when things went wrong. Business Manager (C)³⁸

... the culture is risk averse and that means in a sense that we probably built more than you would do in other circumstances because you give more weight to audit type requirements ... The ability to make effective decisions is a cultural aspect of the Department in that people don't particularly want to (I'm glad you are recording this actually) people don't want to make decision because they don't want the blame attached to them if something goes wrong. Client Project Manager

Empowerment is not enough, there has to be a willingness to make important critical business decisions. This analysis agrees with Morgan's (1997) views that empowered decision making is inhibited by hierarchical cultures, although in this case study it pertains to the former culture. He states, 'the limits of 'empowerment'

³⁸ A different identifier has been attached to this comment to prevent recognition to other quotes with the same reference number, because it is identifiable to a specific person.
are usually quickly felt as people run into the constraints imposed by the existing hierarchy’ (1997, p169). Empowerment focuses on people rather than the process. The emphasis is on who makes the decision rather than the input into the decision to make the correct decision and avoid blame.

Through in-depth consultation with key business users involved, a third issue was revealed. What was seen as reluctance to take decisions was also partly due to a lack of confidence in the New IT System to meet the critical business needs as set out in the EC legislation. Any failure to comply with the fixed EC deadlines, or non-payment of EC monies for schemes under their responsibility results in financial penalties imposed by the EC. This is further discussed in Section 8.2.3.

Analysis concludes that the issue here is one of both cultural and human aspects. There is a conflict between the bureaucratic environment that is supported by hierarchical structures and the need for fast empowered decision-making that is not a part of the organizational culture.

8.2.3 Decision-making - Stakeholder Co-operation / 'Fit for Purpose' Concept

Within the context of the case study there is a relationship between the 9th DSDM principle - stakeholder co-operation and collaboration, the 4th DSDM principle - 'fit for purpose', and the 2nd DSDM principle - the ability to make authoritative decisions. There are many stakeholder groups as previously described in the case study but the group referred to here are the EC, defined as external stakeholders.

It is responsibility of the Department involved to administer the EC grant and subsidy scheme payments within the set payment periods of each scheme. These payment 'windows' are different for each scheme and are completely inflexible.
Thus, the Department is answerable directly to the EC, and any failure to comply with the fixed deadlines, or non-payment of EC monies results in fines and penalties imposed by the EC. This was the major drive behind the business people, where the 'fit for purpose' criteria may not be sufficient to satisfy the EC legislative requirements. This is contrary to the views expressed by Smith and Fingar (2002) who maintain that there is a fundamental requirement for a system to be flexible to changing requirements, and this is certainly the situation with the research case study – having the ability to accommodate EC scheme changes.

A tenet of a RAD-type development approach is the co-operation and collaboration of all stakeholders, however a serious constraint on the project was the inflexibility of the EC as a major stakeholder. This inflexibility imposed rigidity upon the development approach such that it necessitated a high degree of conformity to business needs in order to satisfy the EC legislation, which in turn impacted severely on development deadlines. Continuing the concerns around the disinclination to make decisions, analysis shows that the same business managers also experienced difficulties in signing off development work. It is essential that they meet the payment deadlines and this contributes significantly to their reluctance to make empowered decisions for the new system. Even though they were empowered to make decisions they were not willing to ‘sign off’ development work if it did not completely enable them to meet the requirements of the EC legislation and regulations. Thus conformity to EC legislation and regulations where the ‘fit for purpose’ requirement may reflect almost 100% of business needs meant that the descoping and time boxing elements of a RAD-type approach were initially difficult to achieve. It was the business managers who were initially expecting all business requirements to be met.
Descoping within the timeboxing concept resulted in a reluctance to agree to reduced scope that was deemed by the Supplier's developers as 'fit for purpose' – but which from a business perspective needed to satisfy the EC requirements. The degree of fitness therefore should be a business decision not an IT decision. It is the business managers who say whether the Suppliers are delivering in terms of scope to the EC requirements but there is evidence to suggest that on occasion project development was IT led. Analysis suggests that the 100% of business needs required need not necessarily be system automated, there is provision within CAPM Division for teams of multi-skilled users to attend to exception cases, where claims are not standard or sufficiently generic to go through the Generic Process. However as the project matured, levels of compromise were applied that facilitated project development to delivery dates and business satisfaction.

8.2.4 Decision-making - Conclusions

The ability to make fast authoritative decisions is a key principle of a RAD-type approach, but within the project arena decision-making proved very problematical and can be measured in terms of missed delivery deadlines.

The difficulty in making the transition from the previous bureaucratic behaviour of referring decision-making up the line management hierarchy is attributed to the bureaucratic nature typical of most Government departments, where people perceived themselves to be working within a 'blame culture' environment. Similarly adherence to the former culture of owning business processes hindered decision-making activities in terms of prioritising development activities that caused considerable delay and affected delivery schedules. More consensus about the ability to descope development work in order to meet timeboxed deadlines, and a greater degree of compromise in the earlier stages of the project would have
resulted in a greater ability to meet development schedules. Weak people management experienced in the project arena was reactive rather than proactive and contrary to the identified need for strong people management with a RAD-type approach. Resulting slippage could have been controlled through more direct and focused supervision and stronger influence exerted over the key decision makers. This slippage caused considerable delay that is attributed to the reluctance to make the necessary empowered decisions.

The desire to move away from a risk adverse culture to the more risk taking environment of a RAD-type approach was unsuccessful. It proved difficult to remove the perception of the blame status attached to decision-making activities, people were not willing to make decisions. Empowerment was not enough, there had to be a willingness to make important critical business decisions. – this willingness was lacking. A more effective management of the Business Champions and a greater understanding of the underlying issues of reluctant decision-making could have alleviated this problem. Additionally more consensus about the ability to descope development work in order to meet timeboxed deadlines, and a greater degree of compromise in the earlier stages of the project would have resulted in a greater ability to meet development schedules.

However decision-making was also affected by the need to satisfy an inflexible EC stakeholder where the 'fit for purpose' concept was insufficient, and poor communications that failed to convince people of the removal of the blame culture.

8.3 Testing – System Testing and UAT Testing

Integrated testing throughout the development lifecycle is the 8th principle of the DSDM Consortium principles. It is fundamental to a RAD-type development
approach and involves the end-users. Good testing procedures are essential to ensure that the information system being developed operates as intended, and fulfils both business and user requirements. However the evidence informs that testing procedures were not embedded as expected with RAD, but occurred at the end of the development cycles, and as such resembled more of a ‘mini-waterfall’ approach. This approach took place within a wider iterative development approach as described in the case study.

When schedules were delayed and delivery pressure increased to meet deadlines, testing became the ‘victim’. It was truncated, speeded up or in some cases not completed. These findings correspond with those of Oz (2002) who maintains that testing tends to be the least respected phase in system development. It was user acceptance testing (UAT) in particular that evidence identified as suffering the majority of problems, this issue is consequently analysed below.

8.3.1 UAT Testing

The researcher spent considerable time observing UAT activities and attending Schedule Meetings. Consequently it is this test area that has been analysed in-depth. The UAT team consisted of former DO end-users relocated on the development site for the greater part of the project. They were scheme specific knowledge holders who ‘UAT’d’ the business rules aligned to the business processes against test scripts to verify that specific business requirements had been met. Their limited knowledge across the range of schemes, lack of training on the new system processes and ignorance of the business rules concepts caused initial problems. Observations confirm that they became frustrated and demotivated because they did not always understand what was expected of them. Comments from the UAT team confirmed this:
Another thing ... is we've never been experts in scheme XX, before I came down here I had nothing whatsoever to do with scheme XX, my area of expertise is scheme YY. It's the testing environment we are in if the priority is scheme XX then that's what we do. User Acceptance Team Member (UAT) (7)

It's been more of a learning curve and difficult for us to be able to interpret what they expect ... because they don't always set up the test shots specifically for business rules, it's perhaps something that has happened in live environment and they want us to redo it in the test environment. User Acceptance Team Member (UAT) (4)

Analysis suggests that UAT was further hampered by frequent moving of the testing environment between floors during development due to considerable pressure and demand for workspace from the core development team. This serves to demonstrate a lack of not only planning, but also the level of importance given to their activities.

8.3.2 Testing – Scheduling

A major problem experienced with testing was the unrealistic estimated timescales allocated for UAT. Through observations made and time spent in Scheme Meetings, analysis confirms that timeframes were worked backwards starting from the immoveable deadline of an EC payment window. Put simply, weeks were counted backwards from the payment window deadline. For example X weeks allocated for testing, Y weeks for development, Z weeks for requirements elicitation and so on, in order to arrive at a scheduled start date. Consequently when problems arose with requirements, development, and/or system test, it was UAT testing time that suffered. UAT Testing was in this sense a 'victim'. There was continual underestimation of realistic timeframes for UAT testing activities. It became truncated, it was reduced and squeezed to meet deadlines, or overran by several weeks. This happened in almost every case.
Contingency time or room for slippage was not built into the schedules. When questioned about this situation the most frequent response from managers was that there would be X weeks until the deadline so time was allocated as seen fit. This had a direct effect on the iterative development approach. When problems were encountered considerable rework was often necessary which then needed to be retested. This in turn might throw up other problems causing more rework and more delay and so on. One particular example concerns a scheme where 3 - 4 weeks was scheduled for the UAT testing of both the scanning and acceptance activities of the Generic Process. This was considered to be a simple process with few business rules. In reality due to problems and retesting activities, development was delayed as it took 11 weeks to complete UAT testing. In some cases it meant that when development deadlines were missed, the delivery schedule was unachievable. As a consequence development was halted and that scheme was realigned back onto the Legacy System for payments to be made within EC deadlines. When questioned a member of the Senior Management Team commented:

... we underestimated the amount of testing that was required, we underestimated the number of iterations that application development was going to have to go through and the estimates I was getting at the time, which all seemed perfectly reasonable were not being realised in practice. Senior Manager (4)

Even so, there are two important points to consider. Firstly, within the deadline constraints it is only possible to apply a reasonable timeframe for testing activities in order to achieve the 'fit for purpose' concept. Secondly there was a degree of over confidence in the ability of the Supplier's developers to develop to requirements received from business managers, assuming that those requirements are correct and reasonable. It should also be remembered that because business managers required a high percentage of functionality in order to satisfy crucial EC
requirements and legislation, the 'fit for purpose' concept from a developmental point of view was often not acceptable from a business view point.

This links into the issues experienced with timeboxing and descoping activities. However, in the context of this project where the project community were dealing with unknown factors and uncertainty, there does not appear to be a solution to accurately estimate testing time. As the project progressed and the schemes were rolled over into the subsequent payment year, people were more aware of potential problems, fewer problems arose and the test environment became more manageable.

8.3.3 Testing – Joint Testing Environment

A major concern for both Supplier's developers and the UAT team was the joint testing environment that evolved during the early stages, and continued during the first two years of the project. It was at this time that the project was suffering from both external and internal factors. Additionally a perceived lack of business management involvement, problematic requirements elicitation, ineffective decision-making and missed delivery were all impacting upon the development environment. Thus for reasons of expediency, and in an effort to hit delivery time scales a joint testing environment was created. However, in doing so it created its own issues. Typically system testing is performed prior to UAT but difficulties arose when they were done in parallel. One problem area related to both test teams raising the same Test Incident Reports (TIRs) causing duplication and confusion. A comment made by one member of the UAT team that was echoed by the other members was:
... you find that system test are coming up with TIRs and we come up with TIRs in UAT and sometimes neither of us know which TIRs we've raised so there could be duplications. Business Manager (4)

The criticism of parallel testing is that UAT are testing the system that has not been system tested, and therefore technically is not in a suitable state for UAT. Therefore UAT are raising TIRs such that the developers became reactive to UAT comments about usability when the application had not been system tested. Moreover, there were periods of time when the system test team were active while the UAT were waiting to test, whilst being pressurised by the business managers to complete UAT so that deadlines could be met. This situation was not resolved.

A further significant problem related to simultaneous testing of system test and UAT using the same data. Observations confirm that for system testing there is a need to stop the system, manipulate data, and in some instances it requires manually changing data to test how the system reacts to certain situations. For example, the need to manoeuvre a database into a certain state to do a test, back it up, then run the tests and finally restore it to its original status. This proved difficult when the UAT team were testing in the same environment. Therefore although these measures were aimed to improve release schedule deadlines they were often counter productive.

8.3.4 Testing – Conclusions

There is some question regarding the efficacy of the testing approach that was taken, and the management of the testing resources in relation to whether testing was either successful or effective, both or neither. The difficulties experienced with estimating testing times can be associated with experience and learning,
particularly in a development environment that was both innovative and unfamiliar to the project community.

However, in the context of a RAD-type development approach and the DSDM 9 principles where integrated testing is an essential element of an iterative development approach, there is clear evidence that although testing was both integrated and continual throughout the development process it evolved into more of a 'mini-waterfall' type structure carried out through the UAT activities.

A major problem experienced with testing was the unrealistic estimated timescales allocated for UAT. When development schedules overran, it was UAT that was reduced and squeezed to meet critical delivery dates. Testing became a 'victim', it became truncated. However, people forgot that the test team and Business were not the final element in the equation - that was the end-users out in DOs. The more UAT was squeezed the less chance there was for Business to go out to DOs to train the users. The focus was always on development. The difficulties experienced with estimating testing times can be associated with experience and learning, particularly in a development environment that was both innovative and unfamiliar to the project community.

8.4 Chapter Conclusions

Analysis has focused on requirements elicitation, decision-making and testing issues that are analysed to be co-dependent through the involvement of the business managers in the requirements gathering and decision-making activities. Requirements elicitation activities alien to the bureaucratic nature of the project arena was problematic and JAD workshops initially difficult. Failure to identify levels of complexity resulted in under scoping of project requirements and
consequently the development and delivery schedules were overly optimistic and
difficult to achieve. This further undermined the upstream planning, design and
development activities and necessitated reorganization of release schedules and
resources. Decision-making lacked consensus, was ineffective and caused delays
that impacted upon the iterative and incremental development cycles. People had
difficulty moving away from the former culture of their previous work patterns and
the perception of a blame culture. Additionally as slippage started to occur it was
testing that suffered the most. Testing was untypical of a RAD-type development
approach and resembled more of a 'mini-waterfall' approach within the iterative
development cycles.

The issues analysed and discussed in this chapter are the subject of conference
papers that have been presented and published in the official conference
proceedings. A complete list is presented in Chapter 9, Section 9.5, p348.
CHAPTER NINE  CONCLUSIONS

9 Introduction

This chapter sets out the conclusions drawn from the analysis of the material gathered from the case study research. It revisits and discusses the four research questions and examines how they have been addressed.

The corollary that presents a table drawn from the opinions and views expressed in the literature is compared to, and extended by a similar table derived from the analysis of the research case study of the large and complex RAD-type development project. This comparison allows the researcher to present a model of critical success factors associated with a RAD-type development that extends its utility beyond the small to medium sized projects recorded in the literature. Thus it will address the recognised knowledge gap within current literature, and contribute to the current debate.

The chapter further presents the constraints and limitations of the research study, examines the integrity of, and any influence occasioned by the researcher. Finally it puts for some directions for further research.

9.1 Research Questions Addressed

The research title is ‘The utility of using a RAD-type development approach for a large, complex information system’. The goal of this research study was to identify the issues that impacted and influenced the application of a RAD-type development approach within a large and complex development arena. The
research aim was to analyse the utility of such an approach through the research objectives. To remind the reader these are set out below:

- to determine the current status of RAD through a critical literature review.
- to examine the application of a RAD approach within a large and complex IS project.
- to identify the core components of a RAD-type development approach, and establish their presence within the case study project.
- to determine the issues that impacted upon, and influenced the utility of the development approach within the context of the case study.
- to present a model of critical success factors drawn from this large and complex development adopting a RAD-type development approach.

Four research questions were asked in order to achieve these goals; these are set out below and examined individually:

**Research Question 1:** Which issues are considered to have the most effect upon a RAD-type development?

**Research Question 2:** What are the core features of a RAD-type development approach, and were they applied to the case study project?

**Research Question 3:** What are the major influencing factors that affected the development process of the research case study project?

**Research Question 4:** Were the problems experienced attributable to the RAD-type development approach adopted, or to other aspects of research case study?
9.1.1 Research Question 1 - Which issues are considered to have the most effect upon a RAD-type development?

This question examined the literature from both academic and practitioner perspectives to establish the issues that are considered to be problematic to a RAD-type development approach. The literature shows that both critics and supporters have put forward views and opinions of RAD as a systems development approach. Although there is limited availability of published academic material, there is substantial reporting of small to medium development projects from the commercial sector. However, there is no similar body of literature reporting on, or analysing larger and more complex development environments, and this has led to considerable debate about the utility of a RAD approach across such milieux (Graham 1989, Osborn 1995, McConnell 1996, Elliott 1997, Hirschberg 1998, Jones and King 1998, Boehm 1999, Beynon-Davies et al. 1996, 1999, 2001, Howard 2002).

A comprehensive and systematic review of the existing RAD literature raises three key themes that dominate the debate. Firstly, the size, and complexity of a project, secondly, issues that coalesce under the umbrella of quality, and thirdly the literature puts forward ‘culture’ as a key research theme. More specifically the literature recognises the development factors of user involvement, user expectation, communications, requirements elicitation and decision-making that are subsumed within these three themes and which stimulate the current debate. The following sections present an overview of the three research themes that are then further addressed through the development factors as described above.
9.1.1.1 Project Size and Complexity

Project size and complexity are presented in the literature as areas of concern for a RAD development approach. However, the literature does not present any definitive interpretation of project size or complexity. Consequently the researcher accepted the categorizations applied by the authors. Nevertheless it must be recognised that the majority of the reporting of RAD projects and the associated views and opinions expressed refer to small and medium sized development projects that represents a bias in this area. Therefore RAD is typically associated with small/medium projects that are described as highly interactive, have a clearly defined user group and are not computationally complex. On this premise the tenet that is voiced is that RAD is not suitable for all types of project.

Even though there are also beliefs expressed that the issues of project size and complexity can be addressed through splitting up large systems development into modules that can be delivered by small teams quickly (Martin 1991, Jones and King 1998), an assumption is inferred that problematic areas experienced within small to medium sized projects will exponentially affect larger more complex environments. However as already established there are relatively few examples of such RAD applications to substantiate the conclusions drawn.

9.1.1.2 Issues of Quality

There is considerable debate in the literature surrounding issues of quality associated with RAD. The ad hoc evolution of RAD has resulted in different interpretations and a diversity of beliefs that are fundamental to individual philosophies and perceptions of its application. In others words, people’s views and opinions are influenced by their experience of its utilization i.e. as a framework/approach, as a method, or as rapid development tools and techniques.
This diversity illustrates the issue of quality. Different views are held about the level of RAD’s inherent quality or a perceived lack of quality that are subjective to its utilization. Some authors refer to RAD as ‘quick and dirty’, referring to a perceived conflict between speed and quality that are linked to reduced costs and fast development (Merkert, 1996 Linthicum 1997, Raffoni 2000, Saleh 2001). Others believe that the integrated team oriented development environment together with iterative development and incremental delivery result in higher quality systems that meet the dual needs of both user and business objectives (Graham 1989, Martin 1991, Osborn 1995, Elliott 1997, Maner 1997, Howard 2002). A conclusion to be considered is that quality is subjective and cannot be precisely defined. It has evolved from the traditional interpretation of ‘on time and in budget’, and moved towards the more recent concepts of the system being ‘fit for purpose’ and/or ‘value for money’. For a RAD-type development approach quality is associated with the ‘fit for purpose’ concept, it is one of the 9 DSDM principles, but this issue will remain subjective because of the heterogeneity of different projects and their environments.

9.1.1.3 Cultural Issues

The literature puts forward ‘culture’ as a key research theme. This refers to how an organization’s culture can impact upon larger and more complex project developments. The different forms of culture that were explored in detail represented individual, group, and organizational cultures. Literature suggests that bureaucratic cultures associated with the public sector environment are more problematic for a RAD-type development. The view is that considerable change is necessary for this genre of development approach to be successful (McConnell 1996, DSDM 2001, Hirschberg 1998, Martin 1991, Osborn 1995, Beynon-Davies 1996, 2000, Elliott 1997, Cross 1998, Boehm 1999, Highsmith 2000).
More specifically, a system development approach is tempered by the absolute nature of the organization concerned. A methodology does not necessarily map directly onto an understanding of the organization, its rationality or the context of its users (Coughlan and Macredie 2002). RAD requires people to work collaboratively, in integrated team environments involving fast, confident decision-making. Where these elements are not in place, or not a part of the organization's rationale it necessitates changes in the organization's culture. The literature highlights issues of stakeholder influence, and team culture where employees' attitudes and behaviour impact upon the development process. Jones and King (1998) report on a RAD project where the organizational culture only permitted part time commitment from the user community and was subsequently unsuccessful.

9.1.1.4 User Involvement, User Expectation and Communications

A consensus that is axiomatic from the literature is that user involvement and user commitment are associated with user expectation and user acceptance. These factors are all influenced by the quality and efficacy of the communications experienced within the project environment.

The role of the user has evolved from the early requirements gathering exercises of the former more structured approaches, to a sustained and increased user involvement of the more recent iterative and incremental development approaches such as RAD. Literature posits that it is not only the issues of identifying the correct types of users, but also the degree of their involvement and level of their commitment in a development project, and communications issues that impact upon user expectation and user acceptance. It is these factors that can influence the success of a RAD-type development approach (Martin 1991, McConnell 1996,
Opinions expressed support the view that for large and complex development environments, where it is not practicable to involve all system users, identifying the correct users is crucial to system development - albeit the end-users or the middle managers depending on the degree of their business knowledge, their availability and commitment. If you involve the right users, the right system will be successfully developed (Ljubic and Stefancic 1994). Thus the link is made between user involvement and success of IS development projects (Martin 1991, Hirschheim 1983, Beynon-Davies et al. 1996, 1998). Moreover, user involvement and user expectation and requirements elicitation are also interrelated, they are linked through the use of timeboxing, a critical factor for a RAD-type approach. Timeboxing is linked to both management of, and levels of user expectation. The premise is that if users are involved in JAD sessions during design and development activities they have an awareness of the system being developed and hence a more realistic view of the system and a higher level of acceptance.

The literature recognizes the need for increased user involvement during development projects with particular reference to complex and highly dynamic development arenas. RAD's iterative and incremental cyclic development involving an integrated team culture and JAD workshops, elicits more meaningful feedback, which when combined with prototyping is linked to realistic levels of user expectation, and increased user acceptance that serve to reduce project risk (Alavi 1984, Martin 1991, Avison and Fitzgerald 1995, Beynon-Davies 1996, Elliott 1997 Boehm 1999). Whereas RAD projects can fail where the users involved and team selection are incorrect. If the wrong users are selected, requirements gatherings is affected, the system's scope will be inaccurate, and the system will not meet the
users needs that will affect user expectation and user acceptance (Carmel 1995, Jones and King 1998).

Recent research emphasizes that communications in system development is very much a behavioural process, and is considered to contribute to project risk (Coughlan and Macredie 2002). Effective communications is crucial in system development and seen as a major component of a successful project. It is critical to user understanding à propos changes in working practices, accepting new business processes, and promoting co-operation within the development arena. Productive communications facilitates organization wide support, it raises awareness and understanding that in turn engenders user commitment and user acceptance and manages user expectation (Coughlan and Macredie ibid). Effective communications require an integrated high-level strategy across the project environment involving both internal and external mechanisms. A strategy needs to be applied within a realistic, practicable framework that provides a two-way communications channel for dissemination and feedback of information (Cross 1998, Balogun and Hailey 2004). Communications need to be adequate to convey sufficient understanding, such that it engenders a common interpretation and level of expectation across all project communities. Information itself is not necessarily transparent, it can confuse, deceive and mislead as easily as it clarifies.

Communication for communication’s sake does not result in understanding. The importance of considering the medium used to convey the message is stressed. An appropriate mechanism used needs to be aligned to the intended audience, as the success of communications is dependent on the sender and the receiver.
having a common point of reference, an understanding, such that the meaning becomes a shared experience (Cockburn 2002, Walsham 1997).

9.1.1.5 Requirements Elicitation and Decision-Making

Requirements elicitation impacts across the three research themes, and the literature acknowledges that it is crucial to all development projects. It is the requirements that provide the basis for the design, development and implementation of the system being developed. User focused workshops for requirements elicitation in the design and development processes, as discussed above, is accepted as a more effective way of understanding user needs than the one-to-one scenarios of early requirements gathering exercises favoured by the more traditional development approaches. The general consensus expressed is that requirements are not a discrete entity, but rather data gathering is an open-ended activity necessitating a plurality of user viewpoints to provide a broad perspective of business activities and the information needed to support them. Although highly error prone, the collaborative team focus of requirement elicitation that encompasses both user contexts and organizational factors increases the degree of success. With regard to large and complex systems, the more complete the requirements the more successful the system is in terms of user expectation and user acceptance (Eva 2001, Coughlan and Macredie 2002).

Literature links requirements elicitation and user involvement to the quality issues of RAD development. It is expressed in terms of the ‘fit for purpose’ principle aimed at meeting the dual needs of the users and the business through JAD workshops, iterative prototyping and early user feedback processes. Increased levels of user involvement in requirements elicitation contribute to higher levels of
quality. However, there is some enquiry that JAD is less effective within larger project environments as it proves difficult to facilitate the meetings necessary to reach consensus on decision-making. Additionally it is the iterative and incremental nature that is also associated with a reduced need for specifying business requirements, and linked to decreased levels of quality (Graham 1989, Martin 1991, Osborn 1995, McConnell 1996, Linthicum 1997, Elliott 1997, Maner 1997, Compton 2002). With RAD quality is not seen as an overhead but rather as a result of the best practices and added value through the development process.

Literature also exposes RAD's reliance upon the ability of the users involved to make empowered decisions within tight development schedules without having to resort to higher management for guidance or control. It is the decision-making processes of the JAD scenarios that encapsulate the 'fit for purpose' concept to meet business needs. However, RAD decision-making focuses on the process rather than on the people. The emphasis is on the need for fast decision-making rather than on the input into making the correct decision, and it is this that contributes to the debate concerning the issue of quality. Where organizational culture represents a hierarchical status, decision-making activities are more problematic for a RAD approach.

9.1.2 Research Question 2 - What are the core features of a RAD-type
development approach, and were they applied to the case study project?

This question examined the literature to ascertain the core features of RAD, and examined them in relation to the case study project to determine whether they were applied or not.
9.1.2.1 Core Features of RAD

Literature stresses that RAD is an iterative and incremental development approach that compresses the analysis, design, build and test phases of the SDLC into short, iterative development cycles that provide the flexibility to respond to the increasingly volatile and dynamic nature of development environments. At its origin the core features of RAD were described as user involvement, prototyping, CASE tools, SWAT teams, JAD workshops and timeboxed development (Martin 1991). More recently both academics and practitioners describe RAD through the DSDM 9 principles that are considered to constitute a RAD approach (See table 9.1).

<table>
<thead>
<tr>
<th>DSDM Principles 1994</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Active user involvement</td>
</tr>
<tr>
<td>2. Teams empowered to make decisions</td>
</tr>
<tr>
<td>3. Frequent delivery of products</td>
</tr>
<tr>
<td>4. Fitness for business purpose</td>
</tr>
<tr>
<td>5. Iterative and incremental delivery</td>
</tr>
<tr>
<td>6. Changes are reversible</td>
</tr>
<tr>
<td>7. High-level requirements – base lined</td>
</tr>
<tr>
<td>8. Integrated testing during lifecycle</td>
</tr>
<tr>
<td>9. Stakeholders collaboration &amp; co-operation</td>
</tr>
</tbody>
</table>

Table 9.1 The 9 DSDM Principles

The 9 principles incorporate the essence of the original features put forward by Martin and extend to address some of the evolving development issues that impact heavily on current systems development environments. This responds to an increasing interest in the more human related activities, and a more project management oriented approach. Although RAD’s philosophy continues to focus on iterative and incremental development, it has evolved to become a more business centred development process that focuses strongly on the complete project lifecycle.
9.1.2.2. **Alignment of DSDM 9 Principles to the Case Study Project**

Existing literature identified the core features of a RAD methodology through the DSDM 9 principles. They are not prescriptive, but underpin the DSDM framework and are applied as appropriate to development projects – this is discussed further in research questions three and four. These principles were aligned against the case study project arena across the three year period of the project, to determine whether a RAD-type development approach been applied to the research case study (see table 9.2 below).

<table>
<thead>
<tr>
<th>DSDM Consortium 9 Principles</th>
<th>Research Case Study</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Year 1</td>
</tr>
<tr>
<td>1  Active user involvement</td>
<td>Partial</td>
</tr>
<tr>
<td>2  Teams empowered to make decisions</td>
<td>No</td>
</tr>
<tr>
<td>3  Frequent delivery of products</td>
<td>Yes</td>
</tr>
<tr>
<td>4  Fitness for business purpose</td>
<td>No</td>
</tr>
<tr>
<td>5  Iterative and incremental delivery</td>
<td>No</td>
</tr>
<tr>
<td>6  Changes are reversible</td>
<td>Yes</td>
</tr>
<tr>
<td>7  High-level requirements</td>
<td>Yes</td>
</tr>
<tr>
<td>8  Integrated testing throughout lifecycle</td>
<td>No</td>
</tr>
<tr>
<td>9  Stakeholders collaboration/co-operation</td>
<td>No</td>
</tr>
</tbody>
</table>

*Table 9.2. Alignment of the DSDM 9 Principles to the Case Study Project*

The above table shows that there is significant evidence that 8 of the principles that constitute a RAD process occurred from year 2 of the Project. Investigation and analysis of research data explains that early in the first year the Foot and Mouth crisis had a severe impact on the project. Active user involvement necessary for requirements elicitation and feedback processes became unavailable due to the re-allocation of key business personnel to deal with the crisis and delaying development for about 9 months. Thus there is some question as to the allocation of the right users, the levels of their commitment and their availability. Requirements were based at a high-level at the start of the project, and suffered from lack of the key knowledge holders but subsequently evolved as
the need for more specific details to understand the dependencies between the schemes became evident.

Serious problems were experienced with decision-making due to the former organizational culture of deferring decisions up the management chain such that key decision-makers found it difficult to move away former working patterns. However as the project matured and progressed, people’s attitudes and behaviour began to evolve such that empowered decision-making became evident in the project. This is paralleled by the ‘fitness for purpose’ principle where decision-makers experienced difficulties in signing of development work that was also culturally rooted. These initial problems also affected the iterative and incremental delivery principle where the inability to make fast, empowered decisions meant that the project did not get into the planned early cycles of iteration.

Evidence suggests that the principles of ‘frequent delivery of products’, ‘changes were reversible’ and ‘integrated testing’ were successfully applied. However, initially these may have been more technical than process based due to the project development focussing on the more technical development of the system architecture as the project endured a lack of user input due to factors outside the project’s control.

It was the principle of ‘stakeholder co-operation and collaboration’ that proved the most challenging, it was only partially successful throughout the project duration. Although co-operation was achieved with the majority of stakeholders, a serious constraint on the RAD development process was the inflexibility of the EC as a major stakeholder. Conformity to EC legislation and regulations where fitness for
purpose requires 98% of business needs impacted on the timeboxing concept, and the de-scoping elements of RAD development that proved difficult to achieve.

In conclusion, table 9.2 above clearly demonstrates the progressive application of the DSDM principles during the project duration. It confirms views expressed in the literature that RAD, when utilised for the first time, is not an a priori experience but rather a learning experience that evolves throughout the project lifetime.

9.1.3 Research Question 3: What are the major influencing factors that affected the development process of the research case study project?

The purpose of this research question was to identify the issues that had an important impact upon the development process in relation to the research case study project. Although there were many factors involved and problems experienced, the aim is to focus on those issues that affected the project’s development, and which impacted on the utility of the RAD-type development approach.

Analysis of the case study materials, presented in Chapters 7 and 8, has identified a number of key factors that were significant in this way. Firstly the development factors associated with the system development. In the context of this particular case study these were user involvement, requirements elicitation and user expectation, decision-making and communications. Secondly, factors external to the project environment that impinged upon the development process i.e. Foot and Mouth crisis and inflexibility of stakeholders. Thirdly, the use of the RAD development characteristics i.e. JAD workshops and prototyping. It is not possible to consider these issues in isolation of each other, consequently there is some
overlap in presenting the major influencing factors that affected the development process of the research case study project.

9.1.3.1 User Involvement

User involvement is a comprehensive and influential issue. For this case study user involvement has focused on the importance of selecting the right users, their availability and, subsequently their allocation and commitment to the project.

A major factor that affected the initial stages of the development project was the lack of early participation of the critical business users due to the Foot and Mouth crisis. This event delayed the initial design and development stages of the project because the crisis became the prime focus and key personnel were reassigned to deal with it. Consequently business motivation and commitment were reduced at a time when the foundations of the project were being created. Thus the initial low levels of key knowledge holders and end-user involvement meant that development did not get into the powerful iterative cycles that had been planned. This cascaded throughout the IAD process that had long-term implications during the project development.

An affiliate aspect to user availability was the need to maintain the day-to-day the Business as Usual activities. It was not possible to insulate users from their daily tasks and it was often the case that the key knowledge contributors were the least likely to be available. Therefore the dual needs to maintain Business as Usual activities and, the requirement to interact within the development environment were in conflict. The consequences of this were twofold, firstly it restricted access to the domain specific knowledge crucial to design and development, and secondly it reduced levels of user commitment that had a knock on effect to user
expectation and acceptance as discussed above. This concurs with previous research by Barrow and Mayhew (2000).

The exclusion of the former Department's IT personnel from the project environment is a significant contributory factor to initial project problems. Valuable knowledge of the domain specific IT development issues and experience of agricultural domain information were not utilised. The potential contribution and benefit of crucial business knowledge both explicit and tacit was lost. In the critical early stages of the project too much faith was placed in the Supplier's IT expertise and their capabilities to comprehend the nature of the system and its inherent substance that proved not to be the case. This is clearly demonstrated in problems experienced with requirements elicitation, allocating the wrong people results in ineffective user involvement, inaccurate requirements, false user expectation and contributes to poor communications.

**9.1.3.2 Requirements Elicitation and User Expectation**

Typically, a RAD-type development approach uses JAD workshops to base line system requirements at a high-level. Although JAD workshops were initially successful, requirements gathering subsequently became highly problematic, and had a major influence upon the design and development activities.

User expectation was affected by the reduced levels of user involvement, fuelled by ineffective communications that prevented the users from sustaining a practical view of the system under development. Additionally, the initial pre-defined high-level view of system scope resulted in a disparity of views between the developers and those business managers involved. This also resulted in different levels of expectation experienced, and the initial conflict and the lack of a common vision.

320
Most importantly the high-level view did not accommodate the degree of detail needed to develop the critical business functionality required by the business managers, and to accommodate the EC legislation. This initial under scoping of the project meant that the planned development schedules were optimistic and unrealistic causing project delays that impacted upon the upstream design, development and release schedules. Consequently this lack of understanding, or exposure to the crucial business information early in the project development lifecycle was a causal factor in delayed release schedules and missed deadlines.

Significant critique is attached to the efficacy of the subsequent JAD workshops. They are described by participants as ‘difficult’ in terms of participation and productivity. The unfamiliarity of such workshops within the hierarchical civil service culture resulted in some resistance and a lack of willingness to openly voice opinions in such sessions. As a consequence there was no common vision and therefore no team identity, unity or spirit, there was no ‘we’ and ultimately no trust in the crucial early stages of development. Problems related to organizational rationality and a perceived blame culture prevented consensus in decision-making. Furthermore problems associated with tacit knowledge during requirements elicitation prove difficult to resolve. Inherent issues typical of bureaucratic environments did not facilitate easy sharing of crucial knowledge.

The iterative and incremental nature of RAD is associated with a reduced need for specifying requirements, however for this development project where the planned iterations did not occur, there was a greater need to specify business requirements that was not initially taken on board and consequently was critical to development schedules. An associated impact was the volume of tacit data representing the expertise and experience of the key knowledge holders that was difficult to gather.
due to cultural working patterns that did not facilitate the sharing of crucial knowledge.

A further aspect of requirements elicitation causing concern was the 'fit for purpose' concept of RAD, that was difficult to achieve because business managers sought almost 100% of business needs in order to satisfy the EC legislative requirements and avoid disallowance penalties. This issue highlights the problems experienced with stakeholders. In the context of this research case study the researcher is referring to the external stakeholder of the project, who were inflexible and therefore had an impact upon the development project. Additionally, the strict deadlines tied to legislative demands from the EC also meant that the project had to be reactive to pressure. The most recent radical changes to the CAP schemes imposed by the EC, had a considerable impact that resulted in moving the 'goal posts' of the system being developed.

Once the planned iterations were abandoned, system development became reliant on requirements elicitation and UAT. Contrary to RAD principles, towards the end of the development cycle, UAT was routinely used to verify achievement of functionality, and that business needs had been met. Consequently where errors were evident it required more development effort to amend than it would have needed if viewed in a prototyping stage. There was little evidence of prototyping providing any real benefit to the project.

9.1.3.3 Decision-making

Decision-making was a factor that was more heavily impacted upon by cultural issues than the other areas. Organizational people maintained a loyalty to the former hierarchical line management structure where, working within a perceived
blame oriented risk averse culture, decision-making was deferred up the management chain. Consequently the need to make fast authoritative decision was problematic and considerable delay resulted from the reluctance of business managers to make the timely, empowered decisions that are associated with the risk based environment of a RAD approach.

An understanding of the underlying issues behind the reluctance to decision-making was lacking, and this is evident in the proposed solution of empowering people to make decision that did not resolve the issue.

Consequently, the initial lack of compromise and consensus among the key decision makers to prioritise crucial development work resulted in missed deadlines and delivery schedules. Additionally, the absence of senior management personnel in meetings meant ineffective decision-making went unchallenged, and decisions were not imposed.

However, the high-levels of tension present within the JAD workshops surrounding the decision-making process led to compromises to meet timeboxed deadlines being made by people who did not fully understand the implications of the compromises made. Additionally the aggressive and unrealistic timeframes imposed by inaccurate scoping of business needs was also responsible for pressurising participants’ decision-making.

9.1.3.4 Communications

The inherent vertical line management structure of the hierarchical background consisted of information and authority flows that were based on control and power,
operating within a risk averse environment where the culture did not facilitate the easy sharing of crucial knowledge.

Effective communications are a critical factor in project success, however for this project it is accepted that communications were poor and ineffective despite the efforts made by project management to disseminate and inform through a variety of mechanisms. A high-level strategy was not operating within a practicable framework and resulted in a lack of continuity and an ad hoc approach to communications across the project arena that were essentially one direction – a downwards, one-way communications stream. The Divisional Offices felt particularly affected believing they had no 'voice' and consequently became demotivated and despondent. This had a knock on affect to levels of user expectations and user acceptance.

Problematic communications were not sufficient to achieve their purpose and introduced a level of risk into the project that resulted in a number of key issues that impacted substantially upon the development project, these are set out below:

- Ineffective and poor communications restricted the sharing of knowledge, led to conflict from inadequate data input, and consequently inaccurate requirements.
- Ineffective and poorly managed communications were a factor that contributed to the unrealistically high-levels of user expectation.
- The visibility of information disseminated lacked sufficient transparency and clarity to engender understanding and this created a communications gap between the project team and other organizational personnel.
Despite the variety of communication mechanisms utilised, they were not of sufficient richness, and there was little effective engagement between the important messages and the audience.

Ineffective communications were a key problem for the Suppliers who suffered from a fundamental lack of initial information sharing and feedback from the business managers during the early stages of the project that affected the development schedules.

Issues of the transparency of messages rather than their visibility meant that their design was inadequate to convey sufficient understanding. It confused and misled rather than informed.

The Divisional Offices were particularly affected by poor communications and this created a knowledge gap such that users developed different interpretations of the 'reality' of the system increasing the risk of unrealistic levels of expectation.

9.1.4. Research Question 4: Were the problems experienced attributable to the RAD-type development approach adopted, or to other aspects of the research case study?

This question explores the issues behind the problems experienced to examine their effect and relationship to the development approach. It attempts to ascertain the difference between the areas of a RAD-type development approach that were problematic within a large and complex development environment, and the effect of the environment itself i.e. its culture, on the development approach adopted. Thus there are two perspectives to be considered in relation to the problems experienced: (1) the RAD-type development approach applied and, (2) the organizational culture of case study project that is a consequence of other aspects
present within the case study arena that influenced the development approach adopted.

Once again there is a degree of overlap between those areas that influenced or were affected by the development approach and the development setting.

9.1.4.1 Problems Experienced – Aspects of a RAD-type Development Approach?

This section considers whether the implications of the problems experienced within the context of the research case study were attributable to aspects of the RAD-type development approach.

Although evidence suggests that the RAD-type development approach was problematic within the large and complex case study environment, analysis posits that the problems experienced reflect aspects of organizational and workforce cultures of both the Suppliers and the Department rather than any pure developmental aspects of the RAD-type approach taken. To qualify the above statement it is necessary to make the exception of the Foot and Mouth crisis that represents an external impact that could not have been anticipated and therefore not planned for. The other external factors of relocation and personnel changeovers are essentially project management issues.

It was the relationship between the RAD-type approach and the nature of the organizational culture that resulted in difficulties associated with the development factors. For example the JAD workshops as mechanisms for gathering requirements are recognised as being successful and effective when placed within iterative and incremental development cycles. In the context of the case study it was the inherent bureaucratic culture of the organization and the unfamiliarity of
the JAD sessions coupled with difficulties of collaborative working and team fusion that proved problematic when placed within a RAD-type development approach. Equally it could also be concluded that the base lining of the requirements at a high level, DSDM principle 7, was problematic but this reflects the culture of the outsourced developers rather than a consequence of the RAD approach utilised.

Similarly active user involvement, empowered decision-making, ‘fitness for purpose’, and stakeholders collaboration & co-operation, DSDM principles 1, 2, 4, and 9 respectively, were all problematic. In part due to difficulties caused by the external factors of the Foot and Mouth crisis and the EC as external stakeholders both outside the project’s control, and partly influenced by individual, group and organizational cultures. Likewise, frequently delivery of products, iterative and incremental development, reversibility of changes and integrated testing, DSDM principles 3, 5, 6, and 8, were considerably influenced by the lack of development iterations caused by project delays outside the RAD development process and thus were not purely a result of the ‘mechanics’ of a RAD approach. These conclusions are clearly illustrated through the analysis set out in Chapters 7 and 8.

The literature informs that across large and complex developments a RAD approach requires radical shifts in organizational attitudes and structures, and the mindsets of people (McConnell 1996 Hirschberg 1998). People are required to work in different way, in integrated teams, collaboratively this was difficult to achieve. Without this change many projects failed because the change to new methodologies, methods and techniques do not fit within the organizational culture. Adherence to the former hierarchical culture was particularly manifest in difficulties with requirements elicitation and decision-making.
Difficulties experienced with authorized decision-making necessary for a RAD-type development approach were not resolved through empowerment of key decision makers. Empowerment was not enough, the willingness to make RAD-type decisions was lacking. An emphasis was placed on who makes the decision rather than on the effort necessary to make the correct decision. Hierarchical cultures inhibit empowered decision-making (Morgan 1997) and this was true of the research case study. Constraints imposed by the former working culture and adherence to previous working patterns impacted significantly upon the RAD-type development approach such that the fast, authoritative decision-making necessary was unachievable.

It is also important to examine the other side of RAD, not the approach itself but the efficacy of its application and the issues that impacted upon its utility. This not only refers to the external factors causing delays as mentioned above but also the iterative and incremental development cycles that were affected by inaccurate and impractical development schedules contributing to development delays. As a consequence the planned design and development iteration stages were under scoped and the delivery schedules were both aggressive and overly optimistic and, although also part due to other internal and external factors, the planned iterations stages did not occur. The Suppliers selected the development approach before a realistic understanding of the either the Department's organizational culture and the nature of the system had been established. They did not possess an adequate understanding of the situation therefore problems were experienced with the development approach that were not clearly visible in the planning stages. They had a simplified view of both the proposed system and the development environment such that they were unaware of the degree of complexities involved with the CAP schemes and the related EC legislation. This undermined planning
and affected the upstream design and development activities, which necessitated rescheduling of both release schedules and resources, effectively impacting upon the development approach. Thus the planned and defined infrastructure that incorporated both the technical architecture and the information engineering framework fell short and this reduced the efficacy of the RAD-type development approach applied.

Additionally the developers experienced difficulties with user involvement that led to poorly and inaccurately defined requirements that further were exacerbated by ineffective communications. The complexity of the business needs that were critical to development could not be defined up-front; additionally due to the abandoning of the planned iterative design and development cycles they were not identified resulting in crisis around delivery deadlines. Working to the 'fit for purpose' RAD concept and to a predefined scope, conflict arose between the Suppliers and the business managers who were looking for complete business functionality. This was difficult to resolve and was further fuelled by the lack of consensus and reluctance with final decision-making. This resulted in a disparity between the developers and the business managers, a lack of trust and differing levels of expectation that prevented the developers from achieving the revised timeboxed schedules.

Lack of awareness of the level of complexities involved was in part attributable to insufficient effort by the developers into analysing the business requirements. They initially adhered to what was perceived as a predefined system scope, adopting inflexible attitudes towards the business people that contributed to the conflict within the project environment.
The lack of prototyping was a significant factor in the levels of user expectation and user acceptance of the evolving system. Prototyping is an important principle of a RAD-type approach and more effective prioritising of the technical resource by the Suppliers in the early stages could have facilitated a more realistic employment of prototyping. The constraints of tight and optimistic deadlines, and the availability problems of the key business expertise during the Foot and Mouth crisis resulted in the failure to realise the planned prototyping stage of the development approach. Lack of prototyping meant that it was primarily the mechanisms of system testing and user acceptance testing that were routinely used to verify functionality, and identify whether user and business needs had been met. This was highly problematic due to the mismatch between developer and client expectations, mentioned above.

A joint testing environment and the sustained delays caused by impracticable development and testing schedules meant that training became dependent on the incremental delivery schedule, such that when slippage occurred, training was deferred, fell behind and was perceived as inadequate. This impacted on user involvement, user expectation and ultimately on user acceptance of the New IT System.

The lack of clarity in the initial planning stages, lack of awareness of scope, no joint plan within the development arena, the inability to provide final and speedy decision-making, expectations levels tied to the ‘TO BE’ vision and the reduced commitment of key people all affected the efficacy of a RAD-type development approach for this large, and complex development arena.
Analysis posits that problems experienced within the project development arena can, in some part, be attributed to the inherent cultural influences and constraints of the organization and its bureaucratic nature. This refers to the organizational structure, the human elements of behaviour and attitude, and the management of people within their organizational settings. These factors had a significant impact upon the development project approach that hindered the RAD philosophy. Strong effective management is considered key to the success of RAD. Within a development environment apposite management is central to change control, to be effective a project must be properly managed, and this involves people management. Effective people management necessitates a harmony between the social patterns of the organization and the behaviour of the people being managed (Kester 1997).

The people management concept refers to the managing of the project personnel in terms of their supervision, commitment to, and participation within the development arena, and their acceptance of the new IT system. However, it is evident that people management within the development arena was ineffective in managing the cultural changes necessary for such an approach. Weak people management influenced the development arena, its effect was felt across all the development factors in question, i.e. user involvement, requirements elicitation and user expectation, decision-making and communications.

For cultural change to succeed a change in the people's mindsets is needed. It is necessary to convert, but not replace, the existing implicit knowledge owned by the workers with new explicit knowledge of the new culture. In this way rather than challenging people's ideas, the adding of new possibilities and new ideas
facilitates a transition in mindsets (McConnell 1996, Balogun and Hailey 2004). The shift away from a traditional risk averse culture to that of a more risk based RAD environment was not initially achieved.

The organizational structure and working cultures steeped in traditional hierarchical driven business policies and procedures governed through line management and a perceived blame culture were contrary to, and hindered the fast authoritative decision-making culture necessary for a RAD-type development approach. In accordance with the criteria identified as defining a bureaucratic culture. The Department’s organizational culture can be characterised as a bureaucratic culture (Weber 1964, Wallach 1983, Hofstede 2003.) It is hierarchy driven, highly procedural and risk averse, operating within a regulated, and control oriented environment. Evidence confirms there are clear lines of responsibility and authority, work is highly organized, compartmentalised and systematic (Wallach 1983 p32, Carnell 2003). Power and control are achieved through specified standards of work tied to a high degree of job specialization supported by the hierarchy of status differentials, i.e. clear lines of authority, which has particular relevance to the decision-making problems experienced. The nature of decision-making, who is involved in the process and their degree of involvement are thus culturally rooted. As mentioned previously a highly regulated behaviour produces people who tend to be risk averse. Organizational behaviour emphasizes control over process rather than over outcome, and are therefore culturally based. Additionally, it is the cultural time horizons inherent in the organizational culture that determine the speed with which decisions can be made. These issues affected the development project in terms of attitudes, behaviour and customs in respect of previous working patterns, commitment, expectations and acceptance of the new system.
It is accepted that the levels of user involvement and user availability were adversely affected by external factors; however they were also influenced by internal issues that had a knock on effect on levels of user expectation and acceptance. The culture of the 'one person one job' resulted in the inability to insulate users from their Business as Usual tasks, affecting the availability of key knowledge holders and subsequently reducing their commitment to the project. Consequently the early design and development stages were delayed by the restricted access to the domain specific knowledge at a time when the foundations of the project were being laid that impacted upon requirements elicitation.

Requirements elicitation through JAD workshops was affected by the former bureaucratic culture of siloed and working attitudes and behaviour such that business managers continued to work independently lacking a common vision that resulted in individual interpretations of the project goals, differing business needs and conflict which impacted upon the project's development progress. The lack of people management and of proactive project leadership enabled the organizational people to adhere to their former working attitudes and patterns.

Integrated teams working collaboratively is a tenet of RAD, however people used to silo working patterns found it difficult to absorb the team culture ethos. Joint project teams did not work closely enough, and this was demonstrated through low attendance at workshops and meetings. Particular people management problems previously exampled refer to the difficulty organizational people experienced in collaborative team working practices, the inability to prioritise business needs and the associated decision-making issues which caused missed deadlines and project delays, which it is felt, could have been controlled through stronger management supervision and pressure exerted of those concerned.
Poor and ineffective communications had a significant impact upon the project. The perceived visibility of communications, concealed issues of a lack of transparency such that messages, largely textual, communicated to the project environment provided limited understanding and created a knowledge gap that was not addressed. The dissemination of information was not sufficient to enable understanding by the recipients. The failure to provide a two-way communications channel resulted in a fundamental lack of information sharing and feedback from the project arena. There was little effective engagement between the message and the audience and this introduced a level of risk into the project that was realised through the unrealistic levels of expectation and subsequently user acceptance.

RAD places a strong emphasis on team culture within system development environments (Bayer and Highsmith 1994, McConnell 1996). Initially this proved problematical for the development arena that experienced trouble in attaining the availability of the right users, and subsequently their allocation to the project environment that affected requirements elicitation, decision-making, user expectation and user acceptance.

The inconsistent focus and conflict between the developers and the organizational people did not engender the close working relationships necessary for the team fusion and team dynamics that are central to a successful RAD-type process (Martin 1991, Beynon-Davies 1998, Jones and King 1998, DeMarco and Lister 1987, Willcocks and Feeny 1997, Highsmith 2002).

Literature proposes that factors associated with culture are key to the success or failure of this genre of development approach within large and complex arenas.
Culture has been discussed in terms of inherent organizational cultural constraints, human elements of behaviour and attitude, and the management of people within their organizational settings. Therefore the cultural dimension does not just reflect an organizational context but also involves peoples’ working habits and attitudes (Martin 1991, McConnell 1996, Elliott 1997, Beynon-Davies et al. 1999, 2000, 2002, Jones and King 1998, Markus 2004).

A closer examination of the cultural influences within the project environment suggests that there was actual impact occasioned by their presence. A RAD-type approach necessitates a development culture based on trust. Within large organizations it necessitates moving away from a hierarchical, top-down organizational structure to an empowered, team-oriented development environment (Martin 1991, Bayer and Highsmith 1994, Carnell 2003, Crozier 1964 cited in Hughes 2004). However the case study context is one of former bureaucratic environment driven by structure and a hierarchy that is traditionally risk averse and operating within a perceived legacy blame culture background. There was a need to establish a climate of trust and co-operation, to engender business support and collaborative working within an anti-blame culture. This could have prevented the problems associated with user involvement, requirements elicitation, decision-making, user expectation and acceptance, and communications that impacted upon the project, and incurred the project delays and overrun that were experienced.

As previously stated a methodology does not necessarily map directly onto an understanding of the organization, its rationality or the context of its users (Coughlan and Macredie 2002). Therefore where a cultural mismatch can be identified, the benefits of the development approach chosen will be either lost or unrealised. Although organizational change was achieved through changing the
business processes and procedures, and the working patterns of the organizational people it did not follow that people's individual or group cultures evolved along the same lines and in parallel with those changes. Culture by its very natures evolves; it cannot be changed quickly or achieved overnight.

Although there was senior management recognition of the on-going problems being experienced there was little evidence of proactive leadership within the working development milieu. Despite senior management recognition of the on-going problems within the development environment it became a reactive arena where problems were experienced, and if not resolved, were addressed in an effort to maintain a level of project activity rather than aggressively progress it forward.

Management of project scope was ineffective. A key contributor to development delays and schedule overruns was the lack of an accurate project scope. Schedules can only be effective if they are a result of a properly and adequately defined system scope, this was not the case. It introduced a risk that when not managed realistically resulted in aggressive and overly optimistic development schedules that were unsuccessful. Managing project scope proved problematical, the lack of awareness of the complexities involved meant that the scope was inaccurate.

Evidence clearly demonstrates that during the first part of the project there was a degree of loyalty that remained with the Legacy System. This was fuelled by the adherence of a number of key knowledge holders to the former bureaucratic culture that affected user involvement, requirements elicitation and decision-making activities that could have been controlled through stronger, more direct
supervision of those concerned. The authority exerted was not sufficient or effective enough to handle the impact these issues had on initial development delay.

Difficulties associated with collaborative team working, user availability and communications influenced by cultural aspects all impacted significantly upon the requirements elicitation process. For successful development of large, complex IS there is a dependency on the completeness of the requirements gathered (Coughlan et al. 2003).

9.1.4.3 Model of Critical Success Factors

The review of existing RAD literature identified the development problems associated with RAD from which a table of key issues affecting RAD projects was developed. The aim of this section is to compare and extend that table with analysis from the research case study, and to put forward a model of critical success factors, drawn from an actual large, complex RAD-type project, that have influenced the application of a RAD-type approach, and which could be beneficial for future comparable RAD-type project arenas.

The model will broaden the perception of RAD’s utility beyond the small to medium sized projects recorded in the existing literature and address the recognised knowledge gap, whilst contributing to the current debate.

From analysis of the research case study materials the researcher has clarified the relationship between the development factors impacting upon the RAD-type approach and the cultural aspects experienced (see table 9.3 below).
Table 9.3 Relationships between the research themes and development factors of the case study project  

<table>
<thead>
<tr>
<th>People Management (Group Culture)</th>
<th>Factors Impacting Upon Development (Development Factors)</th>
<th>Cultural Constraints (Organizational Culture)</th>
<th>Humans Factors (Individual Culture)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>User Involvement</td>
<td>Requirement Elicitation</td>
<td>Decision-making</td>
</tr>
<tr>
<td></td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
</tbody>
</table>

The ticks represent a strong relationship and a '?' shows a weak relationship as highlighted in the analysis Chapters 7 and 8. Strong in this context is not a statistical measure but a reflection of the problems experienced. Data analysis revealed that people management and cultural constraints impacted strongly across the development arena with the exception of testing where there was only a weak association. Testing issues were more strongly associated with the human aspects theme that impacted significantly across all the key areas explored.

The above analysis illustrates that a strong emphasis is placed on cultural aspects present within a development arena, and it is these factors that were identified previously and presented in the initial table.

In conclusion the project factors remain unchanged, the type, size and complexity of the system being developed are all factors that need consideration across RAD development arenas. The human issues that represented the most debate in the literature and which involved issues of types of users, their involvement, behaviour and decision-making abilities should also include the full time availability of users involved. More importantly it is the attitudes of the decision makers that is crucial, being empowered to make decisions is not enough there has to be a willingness to make decisions, this is an important inclusion in the Model.
However it is the organizational factors that need the most amendment. The identification of inherent cultural issues present in a development project environment can be critical to a project's success. Top management commitment has always been a RAD tenet but this needs to be more precisely defined to include the visibility of top management within the project arena and more particularly the adoption of a proactive style of project manager leadership. Affiliated to this is the concomitant need for visible people management such that problems being experienced in the project environment attributable to people's attitudes and behaviour can be supervised and guided.

A further issue of importance refers to communications. This research case study had illustrated the importance of effective and transparent communications. It is not sufficient to assume that the visibility of communications is sufficient to engender understanding and knowledge sharing. There is a need to place a communication strategy within a realistic framework to facilitate effective engagement between the message and the audience such that a communications gap is prevented.

Finally, the need for the familiarization of the RAD development approach within the project arena is recognised. Those cases cited as examples of success represent instances where the RAD-type development process had been already familiar to the project community or where the people involved have been sufficiently prepared and primed before project inception.
Table 9.4  RAD – An Model of Critical Success Factors affecting RAD-type Development Projects (amendments are highlighted in red)

Thus from the above conclusions the following Model of Critical Success Factors, set out in table 9.4 above, which is relative to RAD-type project environments, is put forward.

9.1.4.4 Utility of a RAD-type Approach

Literature puts forward the view that RAD is more suited to small and medium development arenas than large complex development projects and suggests that bureaucratic environments are particularly unsuitable. Bureaucracy is put forward as the enemy of speed because the inherent procedural formality hinders the speed of the development process. RAD is believed to be unsuited to highly complex, dynamic business processes that require a more flexible and adaptable development approach. Complexity intensifies uncertainty and this in turn increases the need for knowledge sharing in decision-making. Hierarchical cultures inhibit empowered decision-making that may be unfamiliar and difficult to achieve but which is essential for a RAD-type approach (Martin 1991, Carnell .

However in the context of this research case study it is not the RAD-type approach that has been brought into question but rather the issues that are coalesced under the umbrella of culture. More specifically it is the issues of individual, group and organizational cultures. Analysis concludes that as a development approach there was no direct implication that RAD was not suited. Both the Clients and the Suppliers believe that, although the journey has been difficult, the RAD-type development approach has been successful for this case study project, particularly in light of its evolving and volatile nature. They suggest that had a traditional ‘Waterfall’ development approach been adopted the project would have been cancelled early during second year. The Single Farm Payment project that evolved out of this project is still on-going but expectations of it’s success are high. The problems experienced with the development factors that were further exacerbated by the culture issues are believed to have lessened due to the evolution and familiarization of the development approach through this research case study project.

The conclusion drawn is that a RAD-type development approach that is imported into a hierarchical driven development environment, in the first instance, is difficult to achieve. The shift from risk averse to risk based culture takes time to evolve. Culture by its very nature evolves and cannot be changed quickly. There is a need to establish a climate of trust and co-operation, to engender business support and to instigate an integrated collaborative working that operates within an anti-blame culture. In this way it is possible to manage problems associated with
user involvement, requirements elicitation, decision-making, user expectation and acceptance, and communications.

Therefore for larger and more complex projects special consideration needs to be given to management and control aspects where the inherent formality of organizational structures, in particular bureaucratic cultures, procedures, empowerment issues and communications hinder the RAD philosophy of collaborative and co-operative working and the 'fit for purpose' ethos. This is confirmatory of previous research. (Bates 1995, Boehm 1999, Jones and King 1998, Beynon-Davies 1998¹, 1999, 2001, 2002,

External factors outside a project’s control will continue to present challenges within this genre of public sector development arena. A question that could be raised reflects the recent radical changes to the CAP schemes imposed by the EC. This concerns whether developing systems to meet specific EC requirements is justified, if the EC continues to make fundamental changes to the system processes being developed that may render them obsolete before the new system can be put in place.

The researcher believes that this case study research confirms the utility of adopting a RAD-type development approach within large and complex environments.

9.2 Constraints and Limitations of the Research Study

This section brings together the constraints and limitations that affected this research study, however some of these issues have been previously referred as
appropriate in their specific contexts to provide rationale and clarity in the body of the thesis.

- Problems associated with securing the necessary resources to fund this research project resulted in late entry of the researcher into the project environment. The delay meant that researcher started in early 2002 some 10 months after project initiation. However, due to project extension and overrun the researcher was able to complete a 3 year longitudinal study concluding in 2005.

- There were some initial difficulties in accessing existing secondary documentation stored in organizational databases due to difficulties with official permission levels that took some time to resolve. Linked to this was the problem of identifying the current gatekeepers of knowledge areas as the original authors of existing documentation had left the project environment. This was resolved through contact established outside the project environment.

- The scope of this research project extends to the aims and objectives as set out previously. The primary focus was the investigation into the use of a RAD-type approach for the development of a large and complex information system. However an important issue of RAD associated with high levels of user involvement is the level of usability of the developed system, and although it would be advantageous to include an in-depth usability study this was not possible due to the time restrictions of the researcher and the geographical constraints imposed by the spread of the client base.
• A further problem encountered involved the practicality of conducting the interviews due to the pressure of work the employees were experiencing. However this problem was overcome by introducing flexibility into the interview sessions whereby the researcher would be available on site for a whole day so interviewees could attend when convenient within the timeframe allotted, this proved very successful.

• Additionally as the analysis and review of the existing in-house Project papers progressed the researcher recognised that a positive spin 'before substance' premise occurred in the recording of some events. Although operationally this approach is acceptable to the organization, the researcher realised the importance of taking this into account when examining the official project documentation. Accordingly some allowance was made for the intended audience and the possible motives the author may have had in presenting a positive outcome together with applying triangulation during the data analysis process.

• Due to the limitations of data that could be collected, it was not possible for the researcher to make comments on senior management decisions resulting in contractual changes between the Suppliers and the Department that took place behind closed doors.

• It is recognised that because this piece of research is being funded by the Department that is developing the New IT System, there was potential for influence to be imposed over the analysis and presentation of findings. However this has not been the case and the researcher confirms and assures the reader that there has been no influence exerted over any of, or
input into, the analysis and findings presented in this thesis. The researcher was given a completely free hand and full access to the project environment and people with the exception of restrictions being placed upon the commercial aspects of the project.

- Due to a commercial aspect that is attached to the technical infrastructure and utility of the technology deployed, the researcher was not a party to, or had access to any financial or contractual issues of the project. As a consequence access to any commercially sensitive material was not permitted. Consequently it was not the aim of this piece of research to report on, or evaluate the technical aspects of the project from either a technological viewpoint or to offer criticism of the components used.

9.3 Researcher's Integrity and Influence

Walsham (1995) suggests that with in-depth case study research it is inevitable that researchers will influence those people being researched and/or what is happening within the domain of action. Consequently an area to be clarified refers to whether the researcher had any impact on, or affected the interviewees or the development project in any way such that a bias was introduced. Consequently all interviewees were questioned on this issue.

Consensus posits that the researcher did not in any way influence, or affect the development project in terms of the development process or the product. The majority of those interviewed also reported that they as individuals had not felt influenced or affected by the researcher in respect of their responses. The initial period of orientation and the integration as spectator into the project had enabled the researcher to become accepted as a 'member' of the project environment

345
rather than as an outsider who might affect those being observed and interviewed. However it was noted that some of the senior managers felt that the interview sessions had caused them to reflect retrospectively upon their activities such that, in a subsequent project, they may reconsider their actions.

9.4 Further Research

No research project is totally complete and this research study has raised some interesting areas that the researcher has been unable to pursue but which are worthy areas for further research.

One such area for further research relates to issues of trust between the outsourced suppliers and the organizational people. In the context of this research case study evidence was put forward of an ‘us and them’ attitude within the project arena surrounding the high level system specification set out in the initial contract. It would be interesting to revisit this issue and investigate the significant events that led to some key changes in project attitudes.

A second area of interest that researcher believes is worthy of further attention is that of the level of usability of the New IT System that was outside the focus of this research study. Usability is an important issue and has particular relevance for RAD development arenas that experience cultural difficulties.

Additionally there is some industry keenness surrounding the utility and innovation of the technology and architecture deployed that has attracted some commercial interest. However the researcher was not a party to, or had access to any commercially sensitive material. Consequently due to insufficient data it has not been possible to examine or evaluate the technical aspects of the project from
either a technological viewpoint or of the development components used. Nevertheless it would be interesting to examine this further at a future time once the commercial sensitivity has expired.

A further area of potential research relates to the issue of regulatory bodies external to the project arena. Whether, and how, their regulations and legislation impacts upon the design and development of large scale developments. As discussed in this research case study the EC as external regulators had a significant impact upon the development project.

The findings of this research study pertain to a single case study of the utility of adopting a RAD-type development approach within a large and complex environment. Consequently there is a need for other studies of similar environments to be conducted such that the findings can be confirmed, argued and extended. In particular where this thesis presents a theoretical perspective, drawn from a single case study, further real world research is required in order to substantiate and argue the conclusions drawn. Examples put forward are the issues of the visibility and transparency of communications, and the decision-making issues of authorization and empowerment.

Finally, the New IT System Project evolved into the Single Farm Payment Project that involved the same core team of the Suppliers and their developers, and the organizational people of the Department. Therefore an area of interesting research would be to examine the issues discussed within this thesis in relation to the new project environment with a view to extend, argue and compare the findings put forward.
9.5 Conference Papers Presented and Published in Proceedings

Detailed below is a list of conference papers presented and published in conference proceedings:


PhD Consortium Presentations

- UKAIS 2004 PhD Doctoral Consortium, Glasgow, UK – Presentation of Research Design and Methodology for PhD Research Project.

- ECIS 2004 PhD Doctoral Consortium, Turku, Finland. – Presentation of an Ethnographic Research Approach for PhD Information Systems Research Project.
REFERENCES


