An Interpretive Field Study of Packaged Software Selection Processes

Brian Andrew Light

Information Systems Research Institute
University of Salford, Salford, United Kingdom.

Submitted in Partial Fulfilment of the Requirements of the Degree of Doctor of Philosophy, March 2003.

Table of Contents

List of Tables

List of Figures

Acknowledgements

Many thanks go to Professor Trevor Wood-Harper, and the rest of the staff at the University of Salford who have ‘encouraged’ me to get this done since I joined the group in 1999.

Also, special thanks to my partner, Chris, for being so supportive, for such a long time.

Abstract

Packaged software is pre-built with the intention of licensing it to users in domestic settings and work organisations. This thesis focuses upon the work organisation where packaged software has been characterised as one of the latest 'solutions' to the problems of information systems. The study investigates the packaged software selection process that has, to date, been largely viewed as objective and rational. In contrast, this interpretive study is based on a 2½ year long field study of organisational experiences with packaged software selection at T.Co, a consultancy organisation based in the United Kingdom.

Emerging from the iterative process of case study and action research is an alternative theory of packaged software selection. The research argues that packaged software selection is far from the rationalistic and linear process that previous studies suggest. Instead, the study finds that aspects of the traditional process of selection incorporating the activities of gathering requirements, evaluation and selection based on 'best fit' may or may not take place. Furthermore, even where these aspects occur they may not have equal weight or impact upon implementation and usage as may be expected. This is due to the influence of those multiple realities which originate from the organisational and market environments within which packages are created, selected and used, the lack of homogeneity in organisational contexts and the variously interpreted characteristics of the package in question.

1 Introduction

1.1 Introduction

People in organisations still face difficulties in getting information technology (IT) to do what they want it to do. In recent years, the selection of packaged software over custom development has been viewed as the 'solution' to this problem. In the information systems research community, this shift has been followed by research efforts gaining ground in this area. However, much of this focus has mirrored organisational trends in the adoption of packaged software application types, particularly that for Enterprise Resource Planning (ERP). Whilst this research is valuable, research into application types has overridden specific investigation into the generic nature of packaged software. This is problematic as focussing upon the nuances of application types at the expense of the issues associated with the packaged software element of a study may have broader theoretical implications. Thus theoretically, the favouring of application types over package software has led to a comparative dearth of work on packaged software. To compound the problem, the work that is available, is largely ignored by 'application types' studies, especially those concerned with ERP, the most pervasive. Moreover, it is also clear that the longstanding history of research in respect of custom development is largely absent

from many packaged software related studies. In summary therefore, there is a need to theorize about packaged software and its place within the field of information systems.

Of course, this leaves a large area of potential research that cannot possibly be conducted within one thesis. The literature that is available on packaged software has therefore been used to help scope the nature of the contribution. Earlier packaged software studies have examined the barriers to packaged software adoption (Gross and Ginzberg, 1984) and selection processes (Chau, 1994). However, the results of recent studies, (Pozzebon, 2001; Sawyer, 2001), when compared with the rationalistic assumptions of earlier studies suggest that packaged software selection is much more complex, and hence, is worthy of further study. Moreover, there is Lynne Markus' call for information systems research to examine the different rationales for adopting technologies (Markus, 2000). Thus, this study aims to reconsider the theory of packaged software selection.

In the following section, the context of the research is outlined in terms of the content and importance of the topic, the tradition in packaged software research and the fit of this research with other studies. In the following section the research objective is presented. The chapter concludes with an overview of the organisation of the rest of the thesis.

1.2 The Research Context

This section expands the context of the research outlined in the introduction. It begins with an introduction to packaged software and its treatment to date in order to

demonstrate the motivations for the study. That is, to respond to the idea that packaged software theory is largely ignored in favour of packaged software application types, and that those studies that exist are mostly positivistic (and hence rationalistic) in nature, particularly those concerned with selection - cf. (Anderson, 1990) for example. Then, as the study is concerned with packaged software selection, theories of organisational decision-making are presented. Following this, reference is made to the research on custom development which has already established the futility of rationalistic views of systems development. Thus, the overall aim of this section is to highlight the futility of rationalistic theorizing in relation to packaged software selection and to situate packaged software research within the field of information systems.

1.2.1 An Introduction to Packaged Software

The well documented 'software crisis', borne out of the problems of how software is (and is not) developed and maintained has been argued to have led to applications backlogs in organisations (Pressman, 1997). Moreover, even when developed and implemented, information systems failures are widely reported. It is doubtful that many people in information systems will not have some key information systems failure such as the French Railways (Mitev, 1999) or Taurus (Currie, 1995; Flowers, 1996) with which they are familiar. Recently, combinations of these problems have been conceptualised as legacy information systems problems, resulting in large funded research efforts such as the Systems Engineering for Business Process Change (SEPBC) programme of the United Kingdom (UK) Engineering and Physical Sciences Research Council. Indeed, a multitude of theories, toolkits and cook books to stop cost overruns, ensure success and generally 'make it work' next

time around are available. It was whilst working on the SEBPC project at the Manchester Business School that I became aware of the latest 'solution' - packaged software[1].

Amongst other benefits, packaged software is supposed to speed up the deployment of applications in organisations, tidy up the technical infrastructure, and through outsourcing, provide organisations with a constantly renewable state-of-the-art IT infrastructure (Klepper and Hartog, 1992; PriceWaterhouse, 1996; McKeen et al., 2002). There is broad agreement that packages are pre-written software systems that organisations use instead of developing a system in-house. For example:

- Bansler and Havn (1994) state that generic systems are systems designed and developed for general use, as opposed to custom systems, designed for a specific user or group of users;
- Laudon and Laudon (2000) define software packages as pre-written and pre-coded, therefore eliminating the need for people in organisations to write software programs for certain functions;
- and Kunda and Brooks (2000) suggest that packages carry a sense of getting something that already does the job instead of starting from scratch.

Packages are often also perceived as faster and more cost effective to implement than custom development. It is argued that because packaged software is developed with large customer bases in mind and, is more or less complete software, it can be readily implemented in contrast to custom development (Andersson and Nilsson,

[1] Also known as Commercial-off-the-shelf (COTS) software, standard software, shrink-wrapped software, commercial software, generic systems and application software packages.

1996; Dube, 1998; Lassila and Brancheau, 1999). Carmel (1997) focuses on the commercial perspective of packaged software and suggests that it is sold as a tradable product. Steve Sawyer, who has worked with Carmel in this area, unsurprisingly offers commensurate insights:

> "…packaged software (also known as shrink-wrapped, commercial-off-the-shelf (COTS) and commercial software) means all software sold as a tradable product (purchased from a vendor, distributor or store) for all computer platforms including mainframes, workstations and microcomputers Typically, packaged software is licensed for use, not sold." (Sawyer, 2000: 47)

However, it is important to note that the software is not usually sold, as Carmel and Sawyer would agree. Instead, packaged software tends to be licensed for use. The packaged software product is essentially intellectual property and therefore the vendor retains the ownership of the application and negotiates a licence governing its use with the purchaser (Butler, 1999). Whilst, packaged software may be commercially licensed, there are applications, such as Microsoft Internet Explorer and Netscape Navigator, that are not licensed on a commercial basis although users still enter into a licensing agreement. However, this research is mostly concerned with packaged software which does incur license fees and therefore this aspect is included.

Like most studies, this one focuses upon packaged software used in work organisations. However, it is critically important to point out the significant shift in home computing and the considerable emerging domestic market for packaged software. This includes applications supporting 'Domestic Automation' (For example: Microsoft Money, Email and applications that assist in home improvement such as garden design), 'Entertainment' (Davis et al., 1996) (games for personal

computers and consoles) and 'Learning' (language and encyclopaedias such as Microsoft Encarta). In no way are these categories perfect or exhaustive. These examples merely illustrate that packaged software is pervading domestic settings.

Therefore, packaged software can be seen as software that is pre-built with the intention of it being licensed for use. Although it is problematic to attempt to strictly define the concept – it is necessary to inform the reader of the conceptualisation for the purposes of the study at hand. The point of this definition is that it recognises that packages are products that require licensing for use, but it is inclusive enough to cover software which is not commercially licensed or used in work organisations.

1.2.2 The Research Tradition in Packaged Software

It has been suggested that the packaged software industry traces its roots to the 1968 United States' (US) Justice Department's decision to force International Business Machines (IBM) to unbundle hardware and software (Carmel, 1997). In the late 1970s and early 1980s they were primarily viewed as a way to 'deal' with the applications backlog (Gremillion, 1982). Since then, people in organisations have used packages to varying extents although this has mostly been in smaller organisations (Markus and Tanis, 2000). In the mid 1990s, ERP packages became an incredibly popular choice for larger organisations and today packages are pervasive (Light and Holland, 2000; OECD, 2002). However, despite the phenomenal uptake of packages by organisations, until around 1998 they had received scant attention, in contrast to custom developments, by the information systems research community. Even now, most of the research is concerned with

packaged software application types rather than packaged software per se (Esteves and Pastor, 2001; Westrup, 2002).

This growth and turbulence in the market is reflected in the interest of information systems researchers. It has been reported that during the period 1997-2000, 189 papers related to the subject of ERP were published (Esteves and Pastor, 2001). Panels on the topic were held at key information systems conferences such as the Americas Conference on Information Systems, the European Conference on Information Systems, the International Conference on Information Systems and the Australian Conference on Information Systems. However, it is again interesting to note that the focus is upon a particular application type rather that packaged software per-se. Indeed, Westrup (2002) argues that much ERP research has followed the organisational fashion for this application type.

Moreover, the information systems community still primarily focuses upon custom software, despite shifts in development strategies in work organisations. Very few courses on packaged software run on undergraduate and postgraduate degree programmes, and where they do, these centre on application types, usually with the sponsorship of the vendor[2]. ERP vendors such as SAP and Oracle have been relatively quick to approach universities to set up joint initiatives. Despite undertaking research into ERP packages, myself, for some years I am still unclear how this happens. This is another selection process requiring investigation. Certainly, ERP as an area of research, has expanded dramatically since 1997 – is it that academics have approached vendors, or vice versa? This may not be important, but it appears that a pre-occupation with this application type has led to distinct

[2] See SAP's University Alliance Programme.

neglect of the development of theory about packaged software which generally has much more longevity than any particular application type. There are aspects of ERP packages that are worthy of study, as is true of other package application types such as Customer Relationship Management (CRM) and groupware. Yet, the theoretical issues that may have broader consequences are generally being ignored. This grounds my motivation for theorizing about packaged software selection.

If an organisation is buying a standardised pre-built product that contains finite functionality then the selection process becomes very important. Thus, it would be expected that throughout the selection process, organisational members would evaluate products to ensure they chose one that met their requirements. To put is another way, it would be silly to buy a Mini motor car if you have 10 kids! There are several studies of packaged software selection process (see Table 3.2 later) but they tend to treat selection in an objective fashion as in the example of the Mini. The majority of studies present the process as one that can be carefully orchestrated and executed to achieve a successful result. The underlying assumption is that those involved in the process have an equal stake and share common objectives that will lead to a sensible decision. However, it is clear that organisational decision making is not so straight forward, and the same has been recognised in the custom development of information systems.

As a result of 'passing over' packaged software in favour of the 'glamour' of application types, valuable lessons that will enable the furtherance of theory in information systems are being ignored. This is because packages, and particularly application types, have been sensationalised and held up as remarkably different

from what has gone before (cf. (Newell et al., 2001; Westrup, 2002)). Instead, it would be preferable to view packages as information system that may share similarities and differences when compared with custom developed systems. Moreover, theories of organisational decision making and custom development are widely acknowledged as subjective as outlined in the next two sections. Why therefore, should packaged software selection be any different? Recognition of prior work in the area, rather than attempts to jump on the latest bandwagon are incredibly important if the field of information systems is to avoid labels such as 'a fragmented adhocracy'. Banville and Landry (1992) afford information systems this label on a number of fronts, but particularly because they perceive research as personal and weakly coordinated in the field and where researchers can gain a reputation by contributing to specific groups – ERP packages for example.

1.2.3 Decision Making in Organisations

Decision making, the process of choosing between alternatives, is based on the idea that if there are no alternatives, there is no decision to make. Decision making theories fall broadly into two camps, normative and descriptive (Rollinson and Broadfield, 2002). Normative models emphasise rationality and assert how decisions should be made whereas descriptive models set out to explain how decisions are made (Jennings and Wattam, 1994). Normative models assume that decision makers:

- have a sound basis of knowledge available to them, so they can know the alternatives and successfully predict a variety of outcomes;
- have excellent judgement so they can rank the outcomes in terms of value;
- are rational in the way they make decisions (Teale et al., 2003).

However, there is a great deal of research which questions rational models of decision making (Brunsson, 2000) . For example, it is rarely the case that decision makers have complete information about alternatives and the use of intuition is more prevalent that may be reported (Kahneman and Tversky, 1983; Jennings et al., 1994; Yiannis et al., 2000). Moreover, preferences are seldom rigid and may change in the light of experience (Singh, 1986). Also, rational models ignore the contexts within which decisions are made, in particular group and organisational influences in this respect (Lye, 1994).

Interestingly, even in the light of these critiques, packaged software selection studies to date broadly conform to the rational choice model. Furthermore, in other areas of information systems where decision making is required, such as systems thinking (Checkland, 1981), systems development (Wood-Harper et al., 1985), strategic information systems (Ciborra and Associates, 2000) and strategy planning (Knights et al., 1997), mechanistic approaches have been shown to be inherently flawed This therefore raises serious questions regarding the appropriateness of viewing packaged software selection in a rationalistic fashion. The general assumption is that a decision making process is linear one. It begins with the identification of a problem that requires solving. Then, people with relevant interest and knowledge are brought together to discuss all the relevant information they can gather and out of this emerges the solution to the problem (Watson, 2002). However, it is useful to acknowledge, and even welcome, contributions to decision making processes that may not be viewed in such a fashion. The second category of theory in relation to decision making, those that offer descriptive models demonstrate this.

A key model in the descriptive tradition is the garbage can model (Cohen et al., 1972). The garbage can model indicates that the process of decision making is far more complex than normative models would suggest. It asserts:

- There is typically more than one problem (others may bring different problems to the table).
- Solutions are rarely created in the decision making process (people come to the table with 'the' solution, and the one that may suit them).
- Who is involved and not involved in the decision making process may affect the outcomes.
- The extent to which those who are involved, are involved, may also affect outcomes. cf. (Teale et al., 2003: 288-289) and (Watson, 2002: 339-340)

The utility of this model is that it allows for a degree of interpretation in respect of what is 'the' rational decision. The garbage can model implies multiple realities (Kaplan and Duchon, 1988) and therefore decision-making becomes less straightforward than the rational model suggests. The process is not linear and is heavily influenced by the social context of the decision making process. Because of these multiple realities, multiple rationales are present which may influence decision making and which may be variously interpreted as rational/irrational, good/bad by those involved (Bullock et al., 1988). Returning to the Mini example – it may seem eminently sensible for the parent to buy the car knowing full well they will not be able to seat all the kids if they want an excuse to go out on their own sometimes. Alternatively, it may be that the kids view the purchase as limiting their chances of getting a lift to school.

These two models imply some form of a decision making process. However, it has been suggested that some decisions will not be implemented and some things will be implemented that have not undergone a decision process (Brunsson, 1994). That is, where there are no alternatives or perhaps more interestingly, where there are perceived to be no alternatives.

In summary the dynamics of packaged software selection processes require further theorization in the light of the models presented here. In particular, there is a need to consider the nature of potential problems and solutions (the organisational situations and packaged software products), how solutions may be formulated and who is involved. These issues have already received significant attention in information systems in relation to custom development and are discussed in the next section.

1.2.4 Rationality in Custom Systems Development

Early systems development efforts were relatively unstructured. Lehman (1980) for instance, argues that in the 1950's the ecstasy of instructing a machine to undertake computations at speed overshadowed the rather dull need for a guiding theory and discipline. However, as organisations came to rely more upon information systems - structured programming, design and analysis methods took on more importance. The rationale for this was that a perfect system would be produced if a logical and structured procedure was followed (Friedman and Cornford, 1989; Fitzgerald, 1996). Underpinning this of course is the belief that a ‘perfect’ system can be produced.

It has been argued that the systems development methodologies can:

- accurately record the requirements of an information system
- provide a systematic method of development so progress can be monitored effectively
- provide an information system within an appropriate time scale and cost
- produce a system that is well documented and easy to maintain
- provide an indication of any changes that need to be made as early in the process of development as possible
- provide a system that is liked by those people who are affected by it (Avison and Fitzgerald, 1995: 11-12)

Similarly, as ideas on systems development and implementation evolved it was felt that the only way of achieving the perfect system was to involve end users (Friedman and Cornford, 1989). Consequently much effort has, and continues to be spent upon rationalising the process of custom systems development and involving end users in an attempt to build the perfect system.

However, despite the process of systems development being historically characterised as inherently rational it has been argued that the custom systems development process (and the decision making which forms part of this) is not necessarily as rational as some structured approaches might suggest (Lewis, 1994). This can be illustrated by considering points in relation to the system developer and the end user.

In respect of the developer, although they are often thought of as the objective expert with a tool kit of logical techniques to make the perfect system, there are other factors as play. Developers may have limited experience or knowledge and their own agenda (Yourdon, 1986; Fitzgerald et al., 2002; Kotamraju, 2002). As Edward Yourdon states:

> "For a majority of computer programmers and analysts now, their jobs simply occupy them from nine to five and enable them to pay the rent and buy two color televisions." (Yourdon, 1986: 207)

This may be at odds with system commissioner's objectives and the organisational situation within which the outcome is to be realised. In terms of end users, they are often assumed to share unitary views of organisations where they are believed to hold an equal stake in systems development when in reality, as with developers, this is not the case (Flynn, 1998). End users will have varying personal or group agendas, levels of interest and degrees of power in systems development efforts (Markus and Bjørn-Andersen, 1987; Quintas, 1994). Clearly then, even in this brief example, there are multiple realities and hitherto potential interpretations of irrationalities in existence. Developers, end users and system commissioners (who may also be end users) are not heterogeneous communities and will variously impact upon decision making processes in custom development. It has for instance, been argued that the developer and the end user cannot be removed from the problem domain (Wood-Harper, 1985). Custom system development is arguably not a rational and linear process. Indeed, it has been characterised, quite some time ago, as whilst holding the potential to be explained by rational motives, also being inherently influenced by politics and power (Markus, 1983; Robey and Markus, 1984).

Even though packaged software may display differences to custom development there are still developers, users, the use of methodologies and of course, hopefully, a resulting information system. Thus, it is difficult to see how packaged software selection theory can be presented as, and taken to be, rational given the lessons from custom development. It appears that theory on packaged software has not kept pace with the changes in custom development theories and therefore it is still rooted in the quest for rationalism evident in systems development in the 1970s.

1.3 Research Objective

This research questions the utility of existing theories of packaged software selection. Existing theories of packaged software selection implicitly or explicitly assume a unitary view of organisations thus ignoring the presence of potentially competing organisational realities. Moreover, these theories assume that it is those in the consumer organisation who will 'control' the process of selection, and therefore they do not adequately account for the influence of others in the market within packaged software is created and selected. Furthermore, packaged software selection is currently conceptualized as a linear, well ordered process.

The objective of this study therefore, is to develop an alternative theory to assist in developing an understanding of packaged software selection. The intention is that the theory will more plausibly account for the multiple realities of the process and the consequent potential 'disruptions' to what is usually described as a well ordered and rational process.

In order to achieve this objective, the study:

- Critically reviews a range of issues associated with existing theories of, and related to, packaged software selection from market and organisational perspectives (chapters 2 and 3).

- Reports and analyses the experiences of people in one organisation in relation to a number of packaged software selection processes (chapters 5 and 6).

- Develops an empirically informed alternative theory of packaged software selection (Chapters 6 and 7).

1.4 Organisation of the Thesis

In order to understand packaged software selection it is necessary to refer to the environment within which this occurs. For the purposes of this study this has been split into the environment within which products are created and sold (the market) and the environment within which they are selected and used (the organisation).

Chapter 2, examines the market within which selection may be performed by reviewing certain aspects of the packaged software industry. This includes specific reference to modes of competition, the process of product development and the role of intermediaries. The major theme of the chapter is that packaged software selection may be influenced and controlled by 'the market'.

Chapter 3 considers the selection of packaged software from the organisational perspective. This chapter begins with a review of the reasons for packaged software selection. Existing theories of packaged software selection process are then reviewed resulting in the presentation of an ideal model. Further review of the literature on packaged software is then used to highlight the problems of the idealised model in terms of its rationalistic assumptions and how it is isolated from the market, the process of implementation and the context of usage. The chapter concludes with a conceptual framework which illustrates how packaged software selection theory is couched in relation to other packaged software research.

Chapter 4 discusses the research process of the study. It begins by examining the research tradition in information systems that is argued to be dominated by positivism. Positivism and the interpretive approach of the study, are then introduced and critiqued. Following this the methodology that was constructed is outlined in terms of methods used – case study and action research, the techniques for data collection and reporting, theory development and the role of the researcher. The approach towards the conduct and evaluation of the research is also outlined.

The story of T.Co. is presented in Chapter 5. It begins in June 2000 and ends in December 2002. This chapter commences with an introduction to T.Co and then offers interpretations of three information systems projects involving packaged software selection. Project 1 began as case study research and then became action research. It involves the selection of office automation, email, operating system and database packages in the context of a migration from an Apple Mac to a Windows environment. Project 2 is based upon action research and is concerned with a

business intelligence package. Project 3 is based upon action research and documents the selection of a package to support client-tracking activities.

Chapter 6 uses the conceptual framework developed from chapters 2 and 3 to organise and analyse the data from the T.Co projects. The analysis focuses upon how the activities involved in the packaged software selection processes differed and were constructed by the differing perspectives of those involved and the nature of the packages in question. The findings from the analysis are used as the basis for the development of the alternative theory of packaged software selection discussed in the final chapter.

Chapter 7 summarises the overall findings of the research, reports the conclusions and offers recommendations for further research in the area. It begins with a review of the research process, specifically in relation to the evaluation criteria outlined in Chapter 4. In doing this, the research objective, methodology and the evaluation criteria are revisited. Following this, the overall findings of the research are presented. The thesis closes with some recommendations for future research and the overall conclusions of the study.

2 A Market Perspective of Packaged Software

2.1 Introduction

In recent years, packaged software has become the preferred form of applications deployment in organisations. Significantly, in contrast to custom development, packaged software is primarily thought of in market terms and this has various implications for the process of selection. In order to begin to illustrate these implications, this chapter examines aspects of the packaged software market. The chapter begins by reviewing the growth and global nature of the market. Following this, the features of competition in the market are discussed with reference to their implications for selection processes. The nature of product development and the issues associated with vendors and implementation partners are also discussed in a similar fashion. This chapter situates the processes of packaged software selection within the environment within which the product is created. It further acts as a basis for the next chapter that focuses upon the organisational aspects of packaged software selection, or the environment within which it will be implemented and used.

2.2 The Packaged Software Market

There is little doubt that the rise of the packaged software industry over the last two decades has been considerable.

- In 1980 the industry was predicted to record sales of US$ 2 billion, growing US$ 8 billion by 1985 but in 1984 a report suggested figures that outstripped the previous estimates stating that sales would reach US$ 10 billion in that year and US$ 30 billion by 1988 (Business Week, 1984).
- In 1991 the world packaged software market stood at US$ 52 billion (Economist, 1994).
- Sales for packaged software rose 280% during the period 1986 to 1995 (Sawyer, 2001).
- In 1995, packaged software sales were said to represent around 20 per cent of the total software market which stood at US$ 77 billion, a 270 per cent increase since 1986 (Leebaert, 1995).
- In 2001 it was estimated that world packaged software markets totalled US$ 196 billion (OECD, 2002).

This increase in the market has been reflected in predictions by consultancy firms. For example, PriceWaterhouse's IT review for 1995/1996 indicated that by 2000, nearly 50 per cent of their respondents would have applications support comprised of over 80 per cent packaged software (PriceWaterhouse, 1996). Even some in-house information systems departments have become packaged software companies such as 'Software Corp' (Dube, 1998).

Much of this rise is arguably due to the adoption of packaged software by larger organisations. Traditionally, it has been suggested that packaged software has been more acceptable in smaller organisations but throughout the 1990s, faced with problems such as the year 2000, packaged software has become more attractive to larger ones (Markus and Tanis, 2000). In particular, this has arguably been driven by two key packaged software application types – ERP and CRM software.

The market for ERP packages for example, has grown spectacularly throughout the 1990s. Between 1995 and 1999 it was estimated that £18.7 billion had been spent on ERP software packages alone. Numerous other predictions have estimated growth in the late 1990s and early into the 2000s – these range from around US$ 31 billion to US$ 66 billion (August, 1999c; August, 1999b; August, 1999d). At the organisational level, implementations can last between one and five years and costs can range from US$ 2 to 130 million (Ross and Vitale, 2000). Couple this with a recent ERP survey which suggested that 62 per cent of organisations had abandoned their 'legacy information systems' after adopting an ERP package (Themistocleous et al., 2001), and the rising market is understandable.

Following quickly behind ERP packages, came CRM packages with similarly astronomical markets and spending. For example, during 1998-1999 Siebel Systems, the CRM packaged software market leader, saw revenue rise by 93 per cent to US$ 790.9 million (Goodley and Bennett, 2000) and the Royal Bank of Scotland committed to roll out CRM packages across the organisation with spending running into to tens of millions of pounds (August, 1999d).

Some of the figures above represent license revenues alone and others include associated software services. These include the consulting, contracting, integration, training, support staffing for hardware and software. This has also grown extensively in recent years. Forrester Research (1998) reports that this segment has grown at 16 per cent per annum for more than a decade and accounted for US$ 180 billion in 1997. Clearly, packaged software has become a critically important feature of contemporary information systems support in organisations and it receives much applause. Statements such as those below are not uncommon.

> ERP packages, including those from SAP AG, People Soft, Oracle, and Baan, have made a tremendous contribution to the world of business (Li, 1999: 31).
>
> When it came to the promise of packaged software, you could say it was an offer we could not refuse. In fact, most of our respondents have said that (PriceWaterhouse, 1996: sic).

However, packaged software is not as unproblematic and successful as it is often described. A number of highly publicised failures have been associated with the ERP packages.

- Hershey, Aero Group and Snap-On have blamed the implementation of ERP packages for negative impacts upon earnings (Scott and Vessey, 2000).
- Cadbury Schweppes implemented plans to fulfil 250 orders where normally they would fulfil 1000 due to the increased complexity and the need to re-train staff post implementation (August, 1999a).
- FoxMeyer drug company's implementation of an ERP package has been argued to have lead to bankruptcy proceedings resulting in litigation against SAP, the software vendor (Bicknell, 1998).

Furthermore, a recent Harvard Business School study findings indicate that 65 percent of executives felt that ERP packaged software had a moderate chance of hurting their business (Cliffe, 1999).

The reported experiences of packaged software are, as would be expected, contradictory and multifarious, but clearly the market is burgeoning and the selection of products in this 'highly charged' predominantly commercial environment offers significant interest.

2.3 Globalisation in the Packaged Software Industry

The packaged software industry is global in nature. In 1991, the US held 78 per cent of the market, with Europe in second place with 16 per cent (Brouthers and van't Kruis, 1997). By 1997, the US still dominated the industry although Japan and Europe had both increased their shares of the market significantly and overall worldwide growth was evident (OECD, 2000).

An important issue in this respect is the question of whether the origin of a packaged software product relates to the inscribed assumptions about the conduct of work that they may display as a result. That is, if software is developed by people in the US – will it be in agreement with the 'ways of life' that exist throughout Europe or Asia? It has been suggested that the origin of the development of the package might be at odds with the country of implementation even though the drive for economies of scale may require a globally acceptable product (Krumbholz et al., 2000). Moreover, Carmel (1997) in attempting to explain the hegemony of the US in the packaged software industry, argues that Japan does not have its creative culture, although he

recognises that the 'American Culture of Software' is an intersection of characteristics including US culture, immigrant culture and that of computer programmers and software professionals. The latter point relating to the culture of the developers refers to 'Mealy's Law'[3] whereby:

> "The eventual structure of the system reflects the structure of the organization that builds the system" (Yourdon, 1986: 81-82).

That is, the product that has been developed apes the organisational structure within which the developers operate (Bloomfield, 1992). Yet, Carmel points out that US culture is in fact US 'cultures'. Moreover, in respect of 'Mealy's Law', the problem is that given the rising complexity of packaged software offerings and the characteristics of organisations in the intended markets, developing organisations may lack the structures in place for developers to mirror. One SAP development team found it necessary to involve customers in the development of its treasury module in an attempt to build a product with high market acceptability (Scott and Kaindl, 2000). These arguments also assume that developers are free to build products in a manner that they wish when in fact they, and the organisations they belong to, are to some extent, constrained by market demands if they wish to continue in business.

Of course, the country of origin of the software, the cultural composition of the development team and their environment may impact upon the eventual product but this is not as simple as first suggested. Certainly, many people I have worked with on SAP implementations have commented on how 'German it is in the way it expects you to work'. What this means is that the globalisation of the packaged software industry has significant implications where the producer aims for

[3] After George Mealy, one of the architects of IBM's OS/360

standardisation to achieve economies of scale across diverse geographical and cultural environments. However, it does not necessarily follow that a product built in the US will turn a corner shop in the UK into Wal-Mart.

Thus, in terms of culture, it is perhaps more useful to consider the global argument in relation to the problems of implementing a generic product that has been targeted at a broad market. Undoubtedly, there will be features of the product that will be engrained with cultures that may be influenced by the origins of production and the ethos of the developing organisation. The issue in relation to selection would appear to be understanding the implications of this for the context within which the product will be used. That is, asking the question of whether the package will fit with the cultures of the organisation within which is to be implemented. Of course the matter is more much more complicated. For instance assumptions are made that those involved in selection are willing and able to find out about this, that cultural changes are not part of the reason for implementation, and that those using the eventual system do not subvert the system to avoid any formal culturally rooted directives in the system.

2.4 Competing in the Market

For vendors, the benefits of selling packaged software derive largely from economies of scale. In order to attain these economies, products have to be standardised so they have broad appeal in the market (Robson, 1997; Fan et al., 2000). Thus, the vendor need only develop the software once and can distribute the cost of development and ongoing maintenance over a large base (Gremillion, 1982; Dube, 1998; Butler, 1999).

Consequently, in order to attain the economies of scale, and thereby increase profit margins, vendors have to implement competitive strategies as in any other business (Porter, 1980; Porter, 1996). This means that the vendor has to determine how they will compete in this environment and within which market/s they wish to operate. From a work organisation perspective, packaged software vendors may, for instance, target a mass market with an application type (Microsoft Windows), a particular customer organisation size (SAP versus Sage for accounting products), or sector (IBM's Corebank) in the banking industry. Vendors may also choose to manage product delivery schedules and consumption patterns, for example by segmenting the market via the use of editions[4]. Editions serve to differentiate among similar low and high-end products particularly via differences in functionality and price. What this essentially means is that a vendor may produce multiple editions of their products for various markets e.g. domestic, business, small business, or sector specific such as education. This segmentation activity may of course also be combined with targeting a particular market. That is, various editions may be made available with a market sector. This is done mainly to allow decisions to be made regarding tradeoffs of cost and functionality. It has been argued that this process requires consideration of:

- The type and size of the software product
- The nature of the market -in particular, the status of buyer and seller power
- The need for market information - vendors may release multiple editions simultaneously to gather information

[4] Note the difference between editions and upgrades. Microsoft launched recently launched two **editions** of Windows XP: Home and Professional. These editions can be seen as **upgrades** of the Windows Me and 2000 (although XP may be viewed by some as a new product).

- The required profit margin - releasing better editions, later may increase the lifecycle of a product (Raghunathan, 2000)

In respect of the process of selection, a way of reading these activities is that they used by the vendor as control mechanisms. As the vendor effectively determines who their market will be, how much they will pay and what they will receive, and when (if ever), the consumer organisations palette of choice is limited.

A number of studies have also indicated that time to market is of competitive importance (Carmel and Becker, 1995; Carmel and Sawyer, 1998). It has even been argued, that in this segment of the software industry, time-to-completion is the most important issue and that future competition will be based upon accelerated release cycles (Carmel, 1995b). However, it has been further proposed by the same author that many packaged software companies have not fully recognised cycle time as a competitive concept and rather, that other pressures dominate such as features, performance, customer service, adherence to standards and price (Carmel, 1995a). In order to improve cycle times it is argued that traditional software development life cycle approaches do not work and that new ones need to evolve (Carmel and Becker, 1995; Dube, 1998). This is particularly important as trust from a customer is reportedly often gained through vendor responsiveness and dependability (Gefen, 2002). Therefore, for vendors to sell products it is not good enough just to have them to market first, they need to release products that are perceived to be of good quality too.

Consequently, whilst vendors may control landscape of selection to a fair degree, they clearly have to respond to market demands to some extent (as further discussed in the next section). Yet, vendors may attempt to circumvent this by engaging in processes of convincing consumer organisations that their products are the 'best' and that they should buy into their vision for the package especially where comparable alternatives exist. Friedman and Cornford (1989) call this 'salesmanship', the idea that users may be taught what is good for them and to do things in a new way. Thus, it is possible to argue that, due to the competitive nature of the packaged software market, the selection process may be directly/indirectly or implicitly/explicitly influenced by vendors who bring their own agendas to 'the table'. This is not generally made explicit in existing theories of selection.

2.5 Product Development

As the market requires customers to buy the products on offer it is widely recommended that there are benefits to be had from including them in product development activities. For instance, it is made clear by Raghunathan (2000) that market information, in the form of customer input was essential. This point is also brought out by Carmel and Becker (1995) in their explanation of the requirements of a process model for packaged software product development and Holmström (2001) in her study of the development of the 'Clusterball' online game. However, the extent and nature of the inclusion of consumers in the development process, and for what purpose, is far from simple. This is of particular importance in relation to selection, because once organisations buy a package they automatically become locked into a vendor's product development trajectory. Consequently, there are

questions surrounding how the product will develop in the future and the extent to which any purchaser will be able to influence that activity.

For example, one study recounts the product enhancement process of the SAP Treasury module which involved a number of European and US organisations (Scott and Kaindl, 2000). The study illustrates just how exclusive the packaged software development process can become. Most of the participating organisations were large global firms from a variety of industries which where pruned by the SAP team. They did this by selecting ones with what they felt to be 'state of the art' knowledge in the area. Moreover, to 'ensure mutual goal alignment', they chose organisations that were willing to change their processes. The authors suggest, that as a result, the chosen individuals felt like members of an 'elite group', were excited to influence the design and to be among the first customers to have the module. The process demonstrates how the packaged software company used customers to improve its product, but in a way that mostly accommodated them rather than their customers. They wanted the knowledge of the market, but not the arguments and complexity of heterogeneous ways of work. Essentially, the organisations in the study submitted to a form of salesmanship.

One example of conflict within the process however, is illustrated whereby there were disagreements regarding the linkage of SAP with other applications in place in the participating organisations. The SAP team was unwilling to write interfaces to support these applications. Instead, the most common applications were catered for and users were told to place a request with their software vendors to work with SAP to become partners. This shows how it very difficult for individual user companies

to influence the design of packages. Indeed, it has been suggested that even where purchasers do get involved, vendors may not view all requirements as relevant (Pozzebon, 2001), especially as they want to maintain a 'generic' product so that they can sell it to as many customers as possible (Bansler and Havn, 1994).

Again, it can be seen how a vendor may reduce complexity through standardisation in order to maximise economies of scale. Moreover, customers may find it necessary to make demands upon the producers of packages. Yet, even where demands were made in a relatively strong fashion, the power of the vendor outweighed that of the customers even though they were large companies.

However, it is generally considered to be more expensive to market products to new customers rather than existing ones (Gronroos, 1994; Buttle, 1995; Payne et al., 1999). Consequently, vendors have to balance their desires for increased profitability via economies of scale with customer demands. Andersson and Nilsson (1996) suggest that price competition is less when selling services such as support and maintenance to existing customers and that contracts for these can generate considerable repeat income. Essentially what this means is that packaged software vendors will have little scope for operating a cost based strategy in relation to the initial sale of licences but they will be more successful in generating profit once the customer is locked-in through maintenance and upgrade services as switching costs increase dramatically. In contrast to the Scott and Kaindl (2000) case, VF Corporation and Reebok were both big enough companies to demand changes in the functionality of SAP to suit their requirements (Brown, 1998; Orenstein, 1998; Stedman, 1999b; Stedman, 1999c; Stedman, 1999a).

Although this is a typical reflection of any competitive environment (Porter, 1980; Porter, 1996), it does raise issues in respect of packaged software selection. Specifically, issues related to the responsiveness of the vendor and the ability of the adopting organisations to influence the development trajectory of a given product may require consideration. This is really no different to custom development where various users at the organisational level deal with the development team in various ways with various outcomes (Quintas, 1994; Flynn and Davarpanah Jazi, 1998). Yet, with packaged software the stakes are higher as this negotiation is undertaken in a market environment where the user base is more diverse and diffuse. Indeed, it has been argued that the market represents a barrier between the developer and the user (Bansler and Havn, 1994). With packaged software, the development process is divided into two processes performed by at least two different organisations. The vendor designs, programs and tests the software, while the customer is responsible for the configuration, the organisational implementation, and the operation of the information system. There may be no other connection between the activities of the two organisations, other than in the sale. In addition, the barrier may be further 'thickened' by the presence of intermediaries as discussed in the next section.

2.6 Vendors and Implementation Partners

2.6.1 Intermediation in the Packaged Software Supply Chain

The packaged software market has been described as a prime target for disintermediation. Giaglis et al. (2002) argue that given the dominance of companies such as Microsoft, intermediaries have struggled to differentiate themselves in order to attract customer. However, the software product market exists in part because of the participation of other non software-vendor participants (Sawyer, 2001). Included

here are consulting groups, system integrators, trainers and other software producers. These intermediaries facilitate the linkage between software purchasers and producers because vendors often minimise their role in implementation. A dominant goal of software vendors is to sell their products, leaving it to others to implement them in consumer organisations (Sawyer, 2001). Intermediaries may even sell the vendor's products on their behalf (sometimes also referred to as vendors, or as resellers). Moreover, these intermediaries may also sell their own services such as consultancy to assist with finding a product and implementing it, support services, customisations and bolt-on products. The software product market therefore gives rise to a software services market. The intermediaries that comprise this market can thus, also affect selection processes.

In contrast to custom approaches where close links between users and developers are considered critical (Flynn and Davarpanah Jazi, 1998; Peppard, 2001), software purchasers and developers use a variety of mediated means to communicate. Some of these links are shown in Table 2.1. Although these links may be necessary for the functioning of the packaged software market, they can represent a barrier between end users and developers. This point clearly made in the previous section where even large organisations did not necessary have all of their requirements satisfied. Moreover, even where direct contact does occur (the preferred kind of link according to Keil and Carmel (1995)), it may be that those involved in the link are unable to convey requirements and interpret them (again as with custom development (Curtis et al., 1988; Flynn, 1998; Lai, 1998). The dynamics of purchaser-vendor links are important considerations in packaged software selection as they give an indication of a consumer organisation's opportunities for influencing the development trajectories

of vendors and how they might deal with other matters, such as support, post purchase.

Table 2.1: Customer-Developer Links in Packaged Software Development
Adapted from: (Keil and Carmel, 1995)

Link	Description
Support line	A unit that helps customers with day-to-day problems
Survey	Textual surveys administered to a sample of customers
User-interface prototyping	Customers are exposed to a demo, or early version, to uncover user-interface issues
Requirements prototyping	Customers are exposed to a demo, or early version to discover system requirements
Interview	One-on-one with end-user; open-ended or semi-structured
Testing	New requirements and feedback stemming from testing
Email/bulletin board	Customers post problems, questions, and suggestions electronically
Usability lab	Specially constructed labs for taping and measuring user subjects at work.
Observational study	Customers are followed for an extended period to learn what they do
Marketing and sales	Representatives meet current and potential customers to obtain requirements
User group	Customer groups convene periodically to discuss software usage and improvements
Trade show	Customers are exposed to mock-up or prototype and asked for feedback
Focus group	A small group of customers are brought together to discuss the software

2.6.2 Relationships Amongst Vendors, Implementation Partners and Purchasers

There is a co-dependent relationship amongst vendors, intermediaries and purchasers. Organisations enter into long-term relationships with packaged software vendors. They do not want to customise software and so they need to become active in user groups, a mechanism by which software buyers collectively try to influence

the vendor's plans for package maintenance and enhancement (Markus and Tanis, 2000). Conversely, the packaged software vendors need customers to buy their products. In addition, it is possible to highlight links between vendors and intermediaries. This is often based on the exchange of information about products, services and market conditions. In cases where an intermediary has contact with customers, a vendor may not be able to offer services such as new versions of products without their assistance. If the vendor bases his range of products/services and products and services from the intermediary then a corresponding dependency relationship arises (Andersson and Nilsson, 1996). Therefore, it is also possible to highlight a co-dependent relationship between vendors and intermediaries. Intermediaries help the consumer organisation in a number of ways and this is their reason for being in business. Unsurprisingly, these relationships amongst purchases, vendors and intermediaries can be fraught with difficulties.

Consultants may offer standard solutions to problems that are very specific to the organisations that are employing them – they may not want, or be able to, grasp specific organisational realities. Consultants may also be viewed as holding too much power, influence and knowledge which may 'walk out of the door' when they do (Skok and Legge, 2001). In one ERP project, the company reported that the documentation provided by the consulting group was not tailored to their needs. For example, a costing invoice was called a 'different outlet' which did not make much sense to their employees (Skok and Legge, 2001). In contrast, Guilbert looked into the use of intermediaries to facilitate their ERP implementation process. However, the management team decided that it was preferable to use internal expertise to enable change management and thus, consultants only used to assist in the technical

configuration of the software (Gibson et al., 1999). However, adopting the solution of training a company's internal staff is only slightly less risky. Employees with experience of packages that are in high demand are often lured away by other organisations who wish to implement the same one (Martin, 1998).

As packaged software is a generic product, popular products may see overwhelming demand for implementation and this may lead to a dearth of support for selection and implementation. The ERP market in the late 1990s is a good example. The ERP market grew so quickly that this led to a shortage of competent consultants (Bingi et al., 1999; Sumner, 2000). Purchasing organisations therefore widely complained about consultants with only a few months training who charge US$ 2,500 a day (Martin, 1998). This further manifested itself in a widespread lack of knowledge about the details of ERP products, particularly where integrations and partner products were concerned (Markus et al., 2000). This issue is not new, nor ERP specific. An earlier packaged software study reported difficulties in engaging users in the implementation process as the development team were perceived by the users as not possessing adequate knowledge of the product in question (Lynch, 1984).

Agenda differences have also been identified. In one study 'Company D' found that their consultants also wanted to get the ERP project completed as quickly as possible. It transpired that they could see a glut of business opportunities on the horizon (Skok and Legge, 2001). Differences in agendas may be further amplified and complicated where multiple vendors are involved due to the existence of

multiple packages and custom components as in 'Best of Breed' implementations (Light et al., 2001; Stefanou, 2001).

Finally, when a consumer organisation has entered into an agreement with a particular software vendor there may be a problem of path dependency. This is closely connected to lock-in, whereby once implemented it becomes very difficult (because of huge switching costs) to select an alternative (Howcroft and Light, 2002). Consumer organisations are effectively committing themselves to upgrading software periodically (and mostly at the behest of the developer) if they hope to avoid major conversion headaches (Markus and Tanis, 2000). Indeed, the hidden costs of support, training, tailoring, maintenance, hardware adjustment, forced upgrade and incremental licensing agreements were bemoaned by 33 per cent of respondents in one study (PriceWaterhouse, 1996). Lock-in and switching costs may also become a problem if the vendor an organisation has purchased from drops out of the market. It has been argued that, due to the relatively low costs of entry into the industry, the financial stability of some vendors is questionable and a cause for real concern (Gross and Ginzberg, 1984).

2.7 Summary and Conclusions

This chapter began by reviewing the rise of the world packaged software market and argued that it is rapidly becoming the favoured mode of information systems support in many work organisations. It then argued on a number of fronts that understanding the packaged software market is important in terms of understanding processes of selection. This is because packaged software products are highly globalised, primarily due to the desire of producers to maximise economies of scale. The

chapter implies that the process of package selection will be influenced by the various parties who comprise the market in order to maintain the economies of scale or for other 'profit related' reasons. In sum, it suggests that selection decisions are ultimately constrained by the 'market'. However, it will be shown in the next chapter that much of what is presented is not incorporated into existing theories of selection. Although this chapter has begun to explain some of the drivers that may influence the selection, the process is usually reported as one that is driven by the consumer organisation. With this in mind, the next chapter focuses upon the selection of packaged software from the organisational perspective.

3 An Organisational Perspective of Packaged Software

3.1 Introduction

In the previous chapter, the market-oriented nature of packaged software was discussed. This chapter builds upon this to consider issues related to the organisational aspects of packaged software selection – the environment within which it will be implemented and used. The chapter begins by reviewing a selection of reasons for packaged software purchase, the 'trigger' for the enactment of the selection process. Then, consideration is given to theories of packaged software selection. This involves a critical examination of the typically reported processes of requirements gathering, package evaluation and the selection decision. This is followed by an examination of the linkage between packaged software selection and its' implementation.

The chapter concludes by drawing together the findings of this and the preceding chapter into a conceptual framework. This framework maps out the landscape of the research. Most importantly, it demonstrates how existing theories of selection do not

take account of the full complexities and nuances of the process. The conceptual framework will be used to guide the research and act as a basis for the development of an alternative theory.

3.2 Reasons for Packaged Software Purchase

Although it has been suggested that time, cost and quality are the three most critical considerations in packaged software purchase (Bocij et al., 2003), the matter is significantly more complex. This section presents an overview of the reasons why packaged software may be purchased. Of course, there are also reasons why packaged software might not be purchased. Indeed, as will be illustrated, the reasons presented here may well be interpreted as reasons against selection. However, the focus is on the reasons for purchase, as this more readily implies that the selection process will be enacted[5]. Also, by necessity, this section is fairly descriptive. The aim is solely to put forward a range of reasons why the packaged software selection process may be enacted. It therefore merely lists the reasons and briefly consider why each exists and, where necessary, why they might be interpreted as reasons for not purchasing packaged software

In summary, it is suggested that some of the reasons why packaged software may be adopted over custom development may be intersubjectively agreed as sensible, and that others may not.

[5] I do realise it is simplistic to think in this fashion. I am quite aware that an organisation may choose to select packaged software even when faced with a raft of reasons not to.

3.2.1 The Desire for Standardisation

Standards ensure that a component 'fits' with the rest of a system (Schmidt and Werle, 1998). Whether it is to fit with 'industry standard' practices (Lassila and Brancheau, 1999) or achieve synergy across national boundaries and product lines (Bingi et al., 1999; Huber et al., 2000), the allure of standardisation is a key reason for the purchase of packaged software. At Dow Corning for instance, it was suggested by a Director in Europe that the SAP product would be a fast and effective way to attain global discipline and integrated common systems (Ross, 1999). The same was true at 'Threads' were the strategy was to implement common processes and IT infrastructures through the SAP package. Business processes were modelled in line with those embedded within the ERP package with the objective being to achieve 90 per cent commonality in global processes (Holland et al., 1998; Holland and Light, 1999a). Even earlier, in 1977, the US Forest Service's top management decided that it would begin using a standardised computer based system, known as ADVENT, for preparing and submitting program plans. The management felt that using ADVENT would improve program planning through standardisation thereby enabling comparisons across units and integrating the process more closely with other forest management activities (Gremillion, 1982). Therefore, not only is the standardisation of packaged software seen as beneficial by the producers, but also the consumers.

The major reason why consuming organisations are attracted to standardised products. As shown in the examples above, packaged software is thought to enable better organisational communication through shared systems and a belief in their ability to engender commonality. Although, it is possible to have some pragmatic

sympathy with this aim, other studies of IT and standardisation suggest that standards cannot resolve problems in communication. For instance, in their study of electronic data interchange (EDI) standards, Damsgaard and Truex III (2000) argue that EDI exchanges are not simply highly standardised technical exchanges. They prefer to see them as part of an ongoing and continuously evolving conversation between complex human organisations. The same argument can be applied to packaged software. Implementing packaged software will not necessarily reconcile communications problems or improve communications capabilities.

Moreover, it cannot be assumed that increased communication capabilities are necessarily a good thing. For example, the selection and implementation of a standardised package, for the purposes of improving communication throughout the organisation at 'Eng Co.' was variously interpreted as good and bad by senior management and sales staff (Light, 2001c). Perhaps more significantly, due to the standardised nature of packaged software, information systems developments in organisations may becoming increasingly homogenised and this has considerable implications for selection decisions in terms of organisational fit.

3.2.2 To 'Overcome' IT Legacy Problems

The problems associated with getting IT to work are often characterised as Legacy Information Systems – old, outdated technologies that are muddled by years of modification, degradation and general lack of attention (Bennett, 1994; Adolph, 1996; Warren, 1999). In the 1980s and 1990s, these problems were high on the agenda for many organisations as soaring costs of maintenance and interference in daily work life through the resultant applications back log became apparent

(Swanson and Beath, 1989; Kelly et al., 1999a; Henderson, 2000b). Yet, strictly speaking, 'legacy' does not embody problematic connotations, nor does it necessarily embody good things. Legacy merely implies something that is handed down and is therefore, in some senses, neutral (Allen, 1990). Therefore, legacy *information systems* should be viewed as handed down information systems[6](Light, 2003a).

No matter how legacy information systems are conceptualised, packaged software has been widely cited as the 'solution' to the problems they may pose (Deloitte and Touche, 1996; Knowles, 1997). Packages are argued to be:

- Well structured and allow for maintenance and future development to be outsourced to a vendor (KPMG, 1998; Butler, 1999; Scheer and Habermann, 2000).
- Easily operated, supported and maintained due to the ability of the implementing organisation to tap into available a skills base for the software (Bingi et al., 1999; Sumner, 2000; Willcocks and Sykes, 2000).
- Well documented and organised (Golland, 1978; Butler, 1999; Scheer and Habermann, 2000).

For example, at Novartis (Bhattacherjee, 2000) and 'Global Petroleum' (Light, 1999), the proliferation of ad hoc systems, minimal attention to maintenance and the lack of interoperability were cited as the reasons for the move to ERP packages. Packages got a further boost as companies began to realise the full impact of the

[6] For a more thorough explanation of this theory see Appendix A, which contains the working paper.

Year 2000 problem, and began to view packages as a solution (Holland et al., 1999c; Markus and Tanis, 2000).

Yet, there are potential difficulties with viewing packages as the 'solution' to legacy information systems problems. At Global Petroleum for instance, the legacy information systems comprised packaged software that had been heavily modified. Packaged software was merely being introduced in an attempt to deal with problems associated with an existing package. To treat packaged software as different to legacy information systems is therefore inherently flawed as they are one in the same. One study highlights the irony of the belief in packaged software as the 'replacement' for legacy information systems. 29 per cent of the respondents said they had lost control of the system development process, which was now governed by the offerings of their package vendors, and 41 per cent stated they were locked-in to the packages they had bought to replace 'legacy' custom built programs (PriceWaterhouse, 1996). The implication of this is that although the selection of packaged software may well help relieve some of the problems of an organisations existing situation, it may also introduce new ones. However, the fact remains that legacy information systems are often characterised as basis of an applications backlog that can be dealt with by the implementation of packaged software. This is discussed further in the next section.

3.2.3 To Deal with an Applications Backlog

It is suggested that, faced with application backlogs due to rising software development costs, the need for rapid deployment of new systems to keep pace with business, and the migration of applications to the desktop through client-server

architectures, organisations have increasingly turned to packaged software to solve a variety of business problems (Knowles, 1997; Li, 1999). Moreover, it has been argued that:

> "Programming developed not just for one user, but for many, can help relieve the bottleneck in user demand for computer systems to process information" (Gremillion, 1982: 51)

As packaged software is pre-built, it has been suggested that information systems managers expect shorter implementation timeframes and faster attainment of project objectives (PriceWaterhouse, 1996; Li, 1999). It has also been argued that packages can be made available without development delay, so the users can implement the package instantly (Heikkila et al., 1991). Indeed, it is further argued that the lengthy lag between a user's requests for a new system and implementation (a supposed feature of custom development) has been replaced by market-based approaches where software vendors can produce new releases faster than consumers can absorb them (Sawyer, 2001). Packages have also been reported as lessening the requirement for extensive resources to be focussed upon maintenance activity as this is essentially outsourced to the vendor (Robson, 1997; Butler, 1999).

However, contrary to the argument that packages may offer speedier implementation times, organisations have to wait for the product to be built (Butler, 1999), and when they have implemented it, they may have to wait for upgrades and maintenance activities to be performed (Gross and Ginzberg, 1984). For example, Dell decided that the deployment cycle for the SAP package would have taken them too long. Their plan, to convert all of the company's information systems to the SAP package, was estimated to require several years to implement and thus the project was abandoned (Fan et al., 2000).

Moreover, when an organisation adopts packaged software, they effectively outsource aspects of their change processes to the developer. That is, if they want to change and this change requires changes to the packaged software, they must wait and hope that the developer does so and in a fashion that suits them (cf. the discussion in respect of the development trajectories of vendors in chapter 2). A good example of this was the rush toward CRM functionality late in the 1990s. Although there were major CRM package companies such as Siebel in the market, some organisations had already implemented ERP packages which did not offer CRM functionality. These organisations needed to decide if they were going to wait for the CRM functionality to be built by their ERP vendor, or if the would stray from the planned single vendor approach and adopt a package such as Siebel (Holland et al., 1999b; Holland and Light, 2001). In an attempt to speed up the time to market, SAP and People Soft acquired or set up strategic alliances with CRM packaged software development companies and then integrated and sold this within their own product. Not only did this move bring the functionality to their customer base quickly, they were also in the position of being able to offer them a product with a 'tried and tested' reputation. In this case – even for the packaged software developer – it was easier to buy rather than build.

The potential implication for selection here, is that although packages are thought to reduce the applications backlog, their implementation may lead to other similar problems.

3.2.4 The Role of Selling

An obvious, but often overlooked issue that needs to be considered in relation to packaged software selection is that of sales activity. Oliver and Romm (2000) suggest that packages could not be a solution to organisational problems unless vendors were selling them. What this means, and as suggested in chapter 2, is that organisations may select packaged software as a result of an approach by a vendor or other implementation intermediary that has actively sought them out to sell them a product. This may include the use of vendor promotions, publications, market surveys, the internet, mailing lists (Kunda and Brooks, 2000). For example 'strong ERP vendor marketing' and 'The right solution and message at the right time ..' have been cited as key reasons for its adoption (Klaus et al., 2000). Similar reasoning has been reported with CRM packages (Light, 2003b). It has been further suggested that demonstrations of packaged software are only effective in showing the 'bells and whistles' and do not fully detail the specifics of the package (Gross and Ginzberg, 1984). This is because the vendor views the demonstration as an opportunity to sell the firm's image as well as the product to a buying group that may consist of representatives from various departments. Therefore, the salesperson has to tailor their pitch to the lowest common denominator. The 'bells and whistles' approach is the most usual one. However, very little research has been undertaken that examines this process when it obviously holds great interest and importance in relation to packaged software selection.

Another, perhaps more subtle form of selling is that undertaken by people in organisations. In scene one of the case of Metallica, (Avital and Vandenbosch, 2000) this process is amply illuminated, although not explicitly discussed. The Head

of Systems Development and the Chief Trainer have to explain to the CEO what the software package they are proposing is, the benefits of implementing it, allay fears about the product in question and the potential problems that might be encountered in implementation such as migration and training issues. Moreover, the process of selling has been argued to be elevated above the product being purchased (Howcroft and Light, 2002).

Therefore, those in organisations may be 'sold' the idea of packaged software and, due to the market orientated nature of this strategy, a particular product. This is evidenced by the various marketing campaigns and literature that can easily be found via the internet, and even on television, in the case of the Windows operating system. The notion of 'internal selling' is of importance here.

Thus, those in an organisation may decide to chose packaged software over custom development because they have been sold the idea. Moreover, this selling activity may continue into the process of selection and beyond.

3.2.5 Cost

Cost is widely suggested as a key advantage of packaged software over custom approaches. In one study, 46 per cent of respondents cited this as a reason for adoption (PriceWaterhouse, 1996) and in another, 72 per cent, (Klepper and Hartog, 1992). It has even been suggested that cost is one of the biggest advantages due to the economies of scale companies can tap into (Chau, 1995). Another advocate of packaged software argues:

> "it should be obvious that buying a package -- unless it is a poor fit or product -- will usually cost less money than creating it in-house. If you are not yet convinced of that statement, my suggestion would be to examine why you don't build your own automobile or make your own shoes. You night also wish to re-read those chapters in Adam Smith's The Wealth of Nations, about division of labor, production and wheel re-invention." (Welke, 1981: 400).

On top of this, the costs of acquisition, implementation and usage of the package are argued to be reliably predictable and lower than for custom developed software (Golland, 1978; Heikkila et al., 1991).

Cost may also be a reason for the selection of one product over another. At Siemens Power Corporation, another part of the Siemens group held more licences for the package that was needed and therefore, as these could be used, this information was added into the reasoning for the decision to implement that product (Hirt and Swanson, 1999).

It is not the intention of the author to engage in a grand discussion about the merits of the economies of scale and costs of packaged software, especially in comparison to custom developments, however it is worth looking at studies that suggest this might not be as straightforward as it first seems. At 'Threads' for example, the overall project was reported to have increased in cost five-fold from original estimates (Holland and Light, 1999a) and it is doubtful that FoxMeyer anticipated the ultimate costs of the acquisition, implementation and usage of SAP, which resulted in bankruptcy proceedings (Bicknell, 1998). These two examples highlight that the decision to implement packaged software is not just about the price of a licence, which may well be suitable for a-priori accounting. There are also costs of implementation which are discussed further in section 3.5.2. Moreover, another very

unpredictable cost is that related to the ongoing maintenance of the software as discussed in section 3.5.3. Packaged software projects may be subject to 'cost over-run' problems normally associated with custom development (Remenyi et al., 1997).

In summary, although packaged software may be widely perceived as less expensive than custom development, there is empirical evidence to suggest that the matter is, again, more complex.

3.2.6 The Perception of a 'Tried and Tested' Product

A significant attraction of packaged software for many organisations is related to the perception of the reduced risks of implementing what is seen as a 'tried and tested solution' in contrast to custom development (Golland, 1978; Chau, 1995). Packages are proffered as designed and tested by the vendor, and in most cases, as having been installed by other organisations allowing for reference site visits by potential purchasers in order to evaluate the product (Heikkila et al., 1991). For example, most packaged software vendor websites contain the lists of high profile company cases that aim to illustrate the potential benefits resulting from the implementation of their product. Thus, it is argued that the conditions for estimating the quality and usefulness of the system and the implications for work content and organisation are much better than in traditional development projects (Bansler and Havn, 1994).

Yet, again there are problems with these assertions. There is the suggestion that packaged software is 'better built' than custom developed software yet studies suggest that there is a lack of rigour in the product development processes of the packaged software product industry (Carmel, 1993; Carmel, 1997).

To compound this problem, it has been argued that much of product development is opaque to most consumer organisations. Consequently, since production and consumption are separated, vendors tend to be evaluated in terms of their products, not their processes (Sawyer, 2001; Howcroft and Light, 2002). Therefore, even if a package has been developed in a less than robust fashion, it is possible that this will be ignored by organisations when they decide to implement packaged software and further overlooked in the selection process. In addition, even if a product is seen to work at a reference site, it does not follow that it will do so in exactly the same fashion in another organisation (Light, 2001c). Finally, the above discussion ignores the fact that there are several instances where custom development has been favoured over packaged software selection because purchasing organisations have been unhappy with the product on offer (Dautermann, 1990; Light et al., 2001). As Quintas (1994) suggests, any form of software development is often a difficult and flawed process in which timescales, resource inputs and product quality cannot be predicted with certainty.

3.2.7 The Availability of a Broader Knowledge and Skills Base

Packaged software is usually produced for a mass market and this inevitably means the wider availability of support in comparison to custom developed software where knowledge of the software is specific to the application. The adoption of packaged software for this reason is evident at the Crosfield, DMC Prints and Nokia organisations where only a few employees were capable of handling the administration and development of their existing custom developed software (Dolmetsch et al., 1998). Furthermore, the benefits of increased familiarity amongst the user population can also be realised. These may include opportunities for

increased intra-organisational and inter-organisational knowledge sharing to enable the speedier, and more successful, deployment of packages (Pan et al., 2001; Newell et al., 2002). Furthermore, many job advertisements now specify the requirement for familiarity with a package, particularly office automation packages such as Microsoft Office. This phenomenon has also witnessed in other forms of packaged software such as SAP. Problems may however, arise if a particular form of package or a specific product become very popular and this may lead to difficulties for those in a consumer organisation being able to obtain the skills they need (as discussed in section 2.6). Therefore, although packaged software may be chosen to 'buy into' a knowledge and skills base, difficulties may arise with popular products. It also follows from this that problems may also emerge if a product is, or becomes, less popular which might mean that the support for the package may be hard to find. Consequently, although the decision to purchase packaged software limits the problem of finding someone who knows about the software (in contrast to custom development which tends to be much more localised) packaged software does share the same problem in terms of acquiring the skills base. For example, the lack of SAP consultants in the late 1990s – early 2000s echoes the reported shortage of Assembly skills in 1994 (Bennett, 1994).

3.2.8 To 'Free up' the information systems Function

Very early in the usage of packaged software, it was recognised that it could lead to the release of information systems personnel to work on other projects (Golland, 1978). Additionally, a recent survey reported that 40 per cent of respondents felt that packaged software would allow for reductions in the in-house development team (PriceWaterhouse, 1996). This is argued to be the case as the consumer organisation

will need to allocate fewer resources to development and maintenance activity because this is outsourced to the vendor (Butler, 1999).

However, it is easy to interpret these findings as a way to slash the need for in-house information systems support. Potentially, there may not be a need for the requirement for a large development work force, however the market-oriented context of packaged software clearly requires the in-house information systems function to perform new tasks and, if customisation is performed, some development work will still be necessary (Light, 2001a).

Thus, the extent of the suggested and perceived possible reductions in the information systems function may be overstated as a reason for packaged software adoption.

3.2.9 To Implement Change and Attain Best Practices

The general opinion is that when those in organisations choose to implement packaged software there will be a need for organisational change (Lucas et al., 1988; Glass, 1998; Light, 2000). Very simply, organisations invariably change the ways that they work in concert with those ways inscribed into the software. Within the context of reasons for adoption therefore, it can be argued that people in organisations may choose to implement a piece of packaged software with the explicit desire to force change, or use the packaged software as the 'excuse' for change (Champy, 1997; Soliman and Youssef, 1998). For example, in one interview I undertook for a piece of ERP research, the manager in question said that the package allowed them to undertake 'Business Process Reengineering (BPR) by

stealth' (Davenport and Short, 1990; Hammer, 1990). The link between packaged software and organisational change is discussed in much greater detail in section 3.5.2.

Somewhat allied to the desire to implement change is the use of packages to adopt best practices. The idea that packaged software is tried and tested has already been discussed. Strongly related to this idea is, that through this process of usage and testing, 'best practices' become inscribed into the software. Again, in the PriceWaterhouse (1996) study, greater functionality and the ability to keep IT current were popular reasons for packaged software adoption. In relation to ERP packages one study also cited 'a desire to fulfil the promise of BPR, to move to a process orientation and recognition of best practices' as key drivers (Klaus et al., 2000). The central theme is that through the purchase of packaged software, it is possible to 'buy into' the best practices, or best processes and functionality, that are written into the software.

This discussion raises a number of questions which will be considered later. These are related to the desire for, and recognition of, the need for change and the appropriateness of adopting so called best practices.

3.2.10 Bravado

The adoption of packaged software for the purposes of trying to impress others is not widely reported in the literature. The main reason the that the author has introduced this idea is that in a paper by Adam and O'Doherty (2000) regarding ERP implementation, in one of the summary tables (number 3) a reason for the adoption

of a particular packaged was cited as "To be able to show the big boys". Unfortunately, although the authors discuss the other reasons for implementation, they ignore one of the most interesting. Neither is this an isolated case. At Dow Corning, a reason for the implementation of a package was because many other chemical companies were implementing it (Ross, 1999). It has also been suggested that a reason for adoption might be that the organisation wants to obtain the kudos of being perceived in the vanguard, and at the 'cutting edge' (Oliver and Romm, 2000).

Whilst bravado is not widely reported in packaged software studies it should not be surprising that this is involved. Custom development and other information systems research is replete with stories that indicate the presence of bravado (Hammer, 1990; Davenport and Stoddard, 1994; Howcroft, 2001). For example, Ferneley and Light (2002) discuss a case of the use of a knowledge management system whereby the 'Peacock Effect' emerged as a result of the users wanting to 'show off'. Such findings clearly further undermine theories of decision making based upon the rational choice model discussed in the introduction to this study.

3.2.11 Policy

Organisations may determine that the principle actions in respect of their information systems support will be rooted in packaged software. That is, the solution may be pre-determined by a parent organisation (KPMG, 1998). For example at Dow Corning there was strong sentiment that the firm should be buying-not developing systems that were large (Ross, 1999). There was a similar situation at Siemens Power Corporation where the use of packaged software was company policy (Hirt and Swanson, 1999). Moreover, the policy was more specific in that it specified a

particular product (SAP) in respect of enterprise-wide information systems support. Thus, there is the potential for policies to interfere with package purchase because of the assumption being made by policy makers that there is a product on the market that will meet organisational requirements and even more so where a particular product has been specified in advance. In essence, the selection process may become invalid.

3.2.12 Functionality

Software characteristics have been highlighted as an important factor in software purchase (Lucas et al., 1988; Chau, 1994). This again, is inextricably linked with the nature of packaged software. If an organisation is 'buying into' packaged software over custom development they will generally not want to get involved in customising the product as this erodes the economies of scale. Those in organisations deciding if they will purchase packaged software will therefore look to for packaged software that contains the functionality they require. However, it has been suggested that software packages seldom, if ever, match user requirements exactly (Weing, 1984). Those in organisations suggest that "Available packages do not adequately reflect my industry" and "My needs are too unique to be adequately represented in available packages" (Gross and Ginzberg, 1984). Furthermore, highly integrated sets of packages (such as ERP) may vary considerably in quality on a module-by-module basis (Andersson and Nilsson, 1996). For example at 'Global Entertainment' single vendor based packages were evaluated. However, they were perceived as being historically built from packages aimed at specific functions and then expanded for enterprise coverage. The IT Director stated that although he felt that enterprise systems were good, his company would have to build around them

(Light et al., 2000). Furthermore, in another study, IT and business managers were reported as perceiving integrated ERP packages as having only one 'best in class' application (Light and Holland, 2000). Therefore, before the selection process is enacted, custom development may be chosen over packages (and vice-versa) depending upon perceptions of what will satisfy the requirements of the organisation.

3.2.13 Summary of Reasons for Packaged Software Purchase

This brief review of the reasons why packaged software may be purchased is not intended to be exhaustive. However, what it does begin to illustrate is that the reasons for selection may not be universally agreed as rational. In particular, it raises to the surface a number of what might be perceived as 'highly irrational' reasons for selection such as bravado. It is also suggested that even reasons which look sensible may rely on incomplete information that leads to beliefs in packages that are unwarranted and misguided. In order to draw out the main thrust of the reasons for (and against) purchase, a summary of this section is presented in Table 3.1

Table 3.1: Summary of Reasons for Packaged Software Purchase

The Desire for Standardisation - Packages are chosen as they are perceived to facilitate integration through commonality. However, standardisation may not be welcomed by some who perceive it's integration capabilities for facilitating transparency and commonality as unhelpful.

To 'Overcome' IT Legacy Problems - In contrast to traditional views of legacy information systems, packages are argued to be well structured, organised, documented and resourced because they are standard, pre-built products. Moreover, the problems of development and maintenance experienced with existing systems may be viewed as being solved, though their implementation. Whilst this may be the case, packages also have the potential to bring new' legacy information systems problems to the organisation.

To Deal with an Applications Backlog - Packages may be chosen over custom development because they can be delivered in a shorter timeframe and therefore assist in reducing any applications backlog. Yet, once a package is implemented, development and maintenance is outsourced an applications backlog may appear again and which is further outside the control of the consumer organisation.

The Role of Selling - Packaged software may be purchased because this development route, and more specifically, a product may be sold to those making the decision. Moreover, these 'selling' activities may originate from the market, the organisation or another 'channel'.

Cost - The cost of custom development is argued to outweigh that of installing a package because it is built in a standard fashion, for mass markets, there are economies of scale. Moreover, package purchase and implementation is suggested to be much more predictable than custom development. However, like custom development, cost overruns are evident.

The Perception of a 'Tried and Tested' Product - Packages are viewed as 'tried and tested' products because they are adopted by a large user base and thus, in contrast to custom developed systems, they are easier to evaluate. Yet, product development processes tend to be ignored in favour of product evaluation even though these have been reported as lacking the rigour of custom developed systems.

The Availability of a Broader Knowledge and Skills Base - As packages are developed for mass markets, it is suggested that they offer a more readily available pool of knowledge and skills to support implementation and usage. This is arguably in contrast to custom development where the pool is very specific. However, the availability of the support, like programming languages for custom development, may depend upon the popularity of the package in question.

To 'Free up' the information systems Function - The pre-built and outsourced nature of packages implies that a large in-house information systems function might not be needed or those that work in one can be allocated other work. However, although packages are pre-built they may require some form of customisation which may not be supported by the vendor. Moreover, packages may not completely replace any existing custom developed or customised packages.

To Implement Change and Attain Best Practices - Packages may be used as an excuse to implement change, as they are pre-built and inscribed with implicit ways of work and functionality. Moreover, these ways of work and functionality may be perceived as 'best in class' further fuelling the desire for change.

Bravado - Packages may be purchased to impress others.

Policy - Packages may be purchased because is it policy not to undertake custom development. Moreover, it may also be policy to adopt a particular product.

Functionality - If there is an available packaged software product on the market that is perceived as satisfying organisational requirements. This may be favoured over custom development.

3.3 Theories of Packaged Software Selection

The selection of packaged software products is fairly well documented and has been highlighted as one of the most critical issues in respect of this mode of systems development (Parr and Shanks, 2000). However, the process of selection is usually theorized in a rationalistic fashion. Most studies describe the process as broadly involving the identification of requirements, the evaluation of packages against requirements, and subsequent purchase on the basis of 'best fit' (Lynch, 1984; Chau, 1994; Nelson et al., 1996; KPMG, 1998). Indeed it has been suggested that buyers will seek to choose packages, that for a given software expenditure, will best satisfy their computing needs (Anderson, 1990). Anderson (1990) further suggests that these decisions will involve three basic components: an assessment and evaluation of the performance attributes of each package, a structure or formulation of evaluations that would facilitate package comparisons and a strategy for choosing amongst alternatives. A summary of the theories of packaged software selection is provided in Table 3.2 and an ideal model, based upon these, is presented in Figure 3.1.

Table 3.2 and Figure 3.1, can be used to further the argument developed so far that packaged software selection is far from rational. For example, it is possible to ask:

- Whose requirements are gathered, and why?
- Whose understanding of the available packages is used?
- How is the package compatibility determined, who is involved?
- How objective can vendor demonstrations be?
- Who decides which package is purchased, and how?

The existing theories do have utility and the model of selection they put forward appears sensible. However, the ideal model oversimplifies the process. In the light

of what has been presented so far in this thesis, it is clear that selection is in no way as clear cut as existing theories suggest. Yet, they are still a useful starting point for understanding packaged software selection, and developing an alternative theory. In the following sections therefore, I will consider the various aspects of the process with the intention of characterising them in a richer fashion.

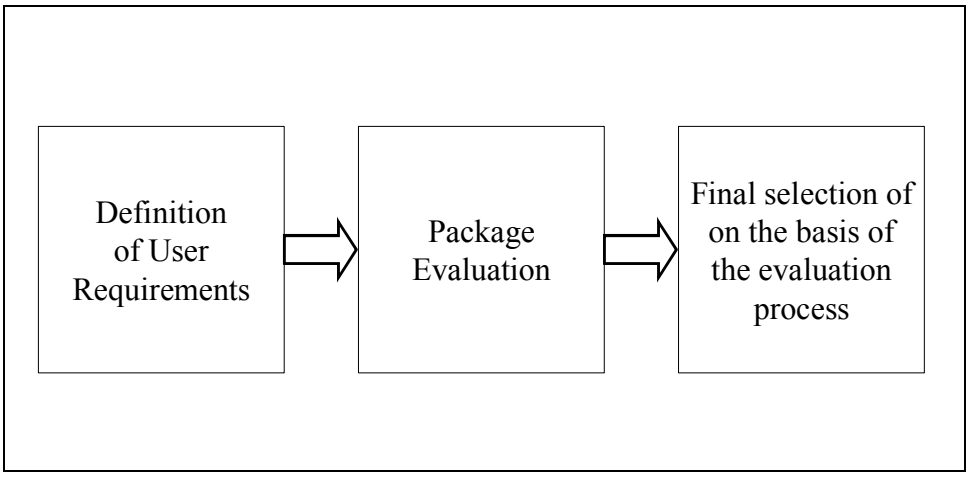


Figure 3.1: An Ideal Model of Packaged Software Selection

Table 3.2: A Summary of Packaged Software Selection Processes

	Definition of User Requirements	Package Evaluation	Final Selection
(Stefanou, 2001)	Organisations should clarify their business vision, requirements and capabilities.	The requirements should be used to evaluate various packages. Packages should be evaluated against each other.	Packages should be selected in the light of the evaluation activity.
(Bansler and Havn, 1994)	The organisation should know its requirement well enough to compare alternatives.	Packages should be evaluated on the basis of organisational and technical requirements and vendor capabilities.	Packages should be selected in the light of evaluation activity.
(Chau, 1995)	Study existing organisational needs and interview some of the future users.	Assess the available packages through vendor representations and comparison with the defined requirements.	Packages should be selected on the basis of maximising the fit between the application and the organisation.
(KPMG, 1998)	Consider 'as is' and 'to be'.		Select package
(Lynch, 1987)	Enough information is required to make the right choice.	Packages should be evaluated on the basis of their functionality.	Packages should be selected that fit with organisational requirements.
(Martin and McClure, 1983)	Identify current and future requirements of the package.	Survey all available packages for the application.	Select the package that most readily fits with organisational requirements.
(Sharland, 1991)	Define requirements.	Evaluate each candidate package using the requirements.	Select the package.
(Nelson et al., 1996)	Identify the company's information processing requirements.	Evaluate packages in respect of their characteristics, company requirements and the capabilities of the company relative to the package characteristics.	Packages should be selected that most readily fit the organisational situation.
(Welke, 1981)	This process is not described in any great detail. The paper suggests that when a requirement for an application arises the best thing to do is categorise the types of software you can buy.	Assess the available packages.	Select a package that most readily meets organisational requirements.

3.4 Packaged Software Selection: Definition of User Requirements

It would be expected, that due to the pre-built nature of packaged software, user requirements gathering would be critical in order to obtain the best fit between the product and the organisational requirements. This is because although end users and information systems personnel may have similar views regarding the ease of use of software packages, their views may differ in respect to the usefulness of software due to a lack of 'organisational knowledge' on the part of information systems staff (Montazemi et al., 1996). Therefore, it has been argued that in packaged software environments, there is a need to need to develop consumer-focused techniques, such as work process analysis, gap-fit analysis, and market analysis in order to assist end users in the selection process (Sawyer, 2001).

This point is further supported by Sherer (1993) who states that packaged software projects embody significant risks associated with function-fit misfits that can only be reduced if users really understand how the proposed system will operate. Similarly, it is argued that in order to get the best technical product, user requirements need to be agreed upon (Kunda and Brooks, 2000)[7]. Therefore, the many proponents of the ideal model of selection would advocate the involvement of users in the requirements gathering and product evaluation activities.

As with custom development though, the extent and purpose of the involvement of users is variable. Views may be sourced only from senior management where their agendas and their perceptions of staff requirements override the requirements of staff. For instance, at one company, the introduction of a Sales Force Automation

[7] This assumes that there is best technical product, that consensus can be achieved and that the organisation knows what it is looking for rather than the selection process being a concurrent educational process as eluded to in section 3.5.2.1.

system was discussed at regional manager level, but did not include the sales staff. In the study, a sales representative is reported as saying that she had been one of three personnel invited to a meeting to discuss possible developments in IT. However, she had arrived to discover that the process was complete and that her role was merely to provide approval for these decisions on behalf of her colleagues (Lloyd and Newell, 1998). Moreover, in extreme circumstances, package adopters will forgo or curtail the analysis of information requirements and completely fall in with those embodied within the packaged software (Markus and Tanis, 2000).

Therefore, there are inconsistencies – user involvement is championed, yet in practice this may not always happen. This is interesting as in a packaged software selection context user involvement could be useful for obtaining an appreciation of package suitability (assuming the users in question have an idea of what is needed) or even on managerially oriented grounds of increasing chances of the acceptability of the package when implemented.

3.5 Packaged Software Selection: Evaluation

Package evaluation may be based upon a number of sources. Word of mouth for example, has been highlighted as invaluable in identifying available packages (Kunda and Brooks, 2000). Others may include publications, market surveys, the internet and mailing lists. However, it has been suggested that these sources may put purchasers at the risk of being enticed by the vendors' marketing people or being locked into a certain way of thinking by looking only at the available packages (Bansler and Havn, 1994). A further alternative might be to experimentation. Indeed, it is suggested that the best way of evaluating packages, is through

experimentation with the operating environment in which the product will be used (Martin and McClure, 1983; Kunda and Brooks, 2000). In an ideal world, this would no doubt be possible, but for most packages is it arguable that this would be too resource intensive. It is difficult to imagine a global company implementing SAP on a trial basis for instance, but a trial of the Netscape Navigator or Opera browser is readily achievable. Indeed, however package evaluation is performed, it is a difficult process because:

- Searching for products is resource intensive, requiring personnel, travel, the development of requests for proposals, gathering and analysing data, development of selection criteria and vendor interviews for instance.
- It may be difficult to find independent guidance. Purchasers may be forced to rely on vendor proposals, presentations and marketing materials.
- If there is a lack of a selection methodology political agendas and/or gut feelings may well dominate rather than well thought out requirements (Hecht, 1997).

What this shows is the wide ranging scope for the process to be shaped by different view points and actions. Marketing documentation for example, however ethical the vendor, will be populated with arguments as to why their product should be chosen. Moreover, even Hecht's advice of having selection criteria in place is no guarantee of objectivity – the criteria themselves are the results of interpretations as is their application. Selection criteria will be specific to organisational context and even individuals, just as with custom development such as those decisions about which requirements are chosen to be satisfied (Markus, 1983; Kling and Iacono, 1984;

Markus and Bjørn-Andersen, 1987). In general however, the major criteria for the evaluation of packaged software centre on the themes of:

- The functionality of the software – currently and in the future.
- The capabilities of the vendor – currently and in the future.

Indeed some of theories of packaged software selection explicitly incorporate these criteria - see (Welke, 1981; Martin and McClure, 1983; Sharland, 1991). Much of the reasoning behind the importance of these themes is concerned with taking a long term view of living with the outcome of the selection process. That is, selecting a product and vendor that meets and will continue to meet the requirements of those in the organisation. Therefore in section 3.5.1, the issues associated with vendors and implementation partners are briefly revisited, adding to those discussed in section 2.6. Following this, section 3.5.2 considers the potential issues associated with functionality of packaged software, section 3.5.3 those associated with customisation and section 3.5.4 those with maintenance and upgrades.

3.5.1 Vendors and Implementation partners

Several studies highlight the importance of consumer organisation views of the vendors of, and implementation partners associated with, particular packaged software products. First, purchase is often motivated by expectations of the direction of vendor products as much as by specific internal needs (Butler, 1999; Sawyer, 2001). That is, a consumer organisation will be interested in the development trajectory of a vendor. Furthermore, the vendor's perceived strength and stability is also a significant consideration (Chau, 1994). In one study it is suggested that:

> "universities are not likely to make an investment in an ERP system that is intended to provide institutional support for many years unless they are convinced that the vendor is totally reliable. Continued enhancement of the ERP product is less likely to be achieved by a supplier of dubious stability." (Oliver and Romm, 2000: 1042).

Reputation, capability, training and support offered, product reputation and whether they are a market leader are further related indicators of the significance of organisational perception of vendor and implementation partner (Kunda and Brooks, 2000). At Siemens, consultants were chosen based on a recommendations from within the company (Hirt and Swanson, 1999) and at Mitek and DMC prints, SAP was chosen as it was perceived as the market leader in ERP packages (Dolmetsch et al., 1998). The whole idea of deciding if a vendor is good or bad is a purely interpretive act. Despite checklists such as those provided by (Sharland, 1991) and (Welke, 1981), the choice of a vendor - as shown above - may be based upon very personal criteria and very little rigour.

3.5.2 Functionality Considerations

A critical part of packaged software selection is that of finding out if the functionality of the package being evaluated is suitable. Indeed, one of the major barriers to the adoption of packages more widely by organisations is a lack of products in the market with appropriate functionality (Deloitte and Touche, 1996). In response, many organisations still adopt custom approaches as with the Nursing Information System at a US hospital, where previously there were attempts at the incorporation of several nationally promoted standard approaches (Dautermann, 1990). However, contrast claims that:

- packaged software provides relatively complete support for standardised business processes (Li, 1999)
- and, it would be expected that it will be some time before regretting any loss of freedom as a result of adopting packages (PriceWaterhouse, 1996),

with the experiences outlined in Table 3.3 and it becomes clear that functionality is a significant issue.

Table 3.3: Examples of Function Miss-Fits

Materials Requirements Planning at American Seating - A shortcoming in the bill of material package was its lack of ability to handle variations of final level assemblies. The implementation of Bill of Material and Product Cost failed to provide an easy way to handle this problem. It was one which the company had to 'crow bar' around (Donoghue, 1983).
Material Requirements Planning at 'Cable' - The ERP software did not sufficiently cater for the notion of length. For example, a customer may want to order a 100-metre length of cable. The ERP software would indicate that the order could be fulfilled stocks totalling 200 metres were held. In fact, physical stock levels comprised 4 x 50 metre lengths of cable (Light, 2001a).
Pricing at 'Home' - The detailed geographical breakdown of Home's market was used to assist pricing based on the distance travelled to deliver the product. The combination of this with other cost elements led to the generation of millions of prices. The customisation was extensive - it applied conditions to the data used to generate the prices lists via custom software developed by an external contractor. The price lists were integrated with the ERP software (Light, 2001a).

3.5.2.1 Packaged software as Organisational Educator

It would be expected that those in organisations would usually need to have some idea of their functional requirements before they begin to assess whether a particular package will fit with these. This is because some required functionality might only be partially met and in others, missing entirely (Butler, 1999). Moreover, the significance of this is increased in the light of the various selling acts that may occur

in the process of selection and procurement, for example, the blurring of packaged software capabilities by vendors as eluded to below:

> "with the huge explosion in the CRM market it is not surprising that the ERP vendors are quick to jump on the bandwagon, touting the benefits of an extended product" (August, 1999b: 38).

However, it is widely reported that software purchasers often do not understand their own requirements and that they may choose a package on the basis of one sales pitch (Butler, 1999). Furthermore, as the software and the systems architecture will be new to an organisation, the business practices that the software supports may also be new and therefore it is suggested that organisations must engage in a process of organisational learning (Scott and Vessey, 2000). The idea that people in organisations may have to learn about the packages they have adopted is touched upon in the literature, interestingly though – this is usually during implementation. For example at Dow Corning, team members who requested modifications to the SAP package they were implementing were often unsure if the need for this had arisen from a deficiency in the package or their ability to learn how the package allowed a process to be performed (Ross, 1999). Adam and O'Doherty (2000) also report that in a few cases the packages adopted led to the development of managerial expectations – they learnt what was possible and became more ambitious in terms of their objectives for the system. These issues do not vary widely from those encountered for some time in custom development. For example, (Flynn, 1998) states that users often only have a vague notion of requirements at the beginning of a project, there may be changes in external environments and initial requirements may have unfeasible implications that are not realised until implementation. So, it is

unexpected that they are being reported, with some surprise, again by those working in the area of packaged software.

The development of knowledge of canonical and non-canonical processes embedded in packages has been cited as a key success and failure consideration in packaged software implementation (Lee and Lee, 2000). What this means is that not only do packages require extensive user training in terms of how to operate the package, there may also be issues associated with the changes the package brings about in the working environment. For example, Cadbury Schweppes implemented plans to fulfil 250 orders where normally they would fulfil 1000 following its ERP implementation. This was reported to be due to an increase in the number of people who are involved in the process who were involved in new ways and therefore needed to re-orient themselves (August, 1999a). A similar story was reported by Skok and Legge (2001). In their study, staff at Company B thought that the reason ERP projects failed was because people were not able to accept standard processes or see the impact of integration on other areas. This issue has been discussed extensively in earlier BPR studies (Hall et al., 1993; Bashein et al., 1994; Martinez, 1995).

This discussion raises the idea that, requirements may not be fully known. Thus, this may have implications for the selection process in terms of evaluating products. It also begins to highlight a link between the selection of a product based upon 'incomplete' user requirements and its implementation and usage in organisations. These issues are not explicitly addressed by the ideal model of selection.

3.5.2.2 Expectations of Functionality Fit

It is argued that a better fit between the packaged software functionality and user requirements leads to successful implementation and usage (Janson and Subramanian, 1995). At a simple level, this seems logical. However, as illustrated in the last section, those in organisations are often uncertain about their precise requirements and therefore it may prove difficult to evaluate a package upon the basis of these 'incomplete' or 'inadequate' requirements. Moreover, even where requirements are fairly well articulated, evaluation can still be problematic as packages may well meet the requirements in an unfamiliar or unacceptable way (Gremillion, 1982). It has been suggested for example, that most package purchasers find that at least 20 per cent of their requirements are missing from the functionality of packages (Scott and Kaindl, 2000). This is potentially no small mismatch either if the missing 20 per cent is that which is perceived to be the most crucial to those in the consumer organisation.

In one study of CRM package implementation, 'ManCo' expected a package to be suitable as they had liked the way they had seen it operating at a reference site that was in a different industry and had different ways of working. In the same study, a package was sold to two organisations as a CRM product yet it the organisations could not use the package to find out who their most profitable customers were[8] (Light, 2001c). Similarly, at the University of Nebraska, the average fit between the implemented SAP packaged was 60 per cent and as low as 30 per cent in some areas yet they bought the package with the expectation of it providing enterprise support (Sieber et al., 2000).

[8] Ironically, CRM is supposed to assist organisations to target their most profitable customers.

This discussion implies that even though it has been argued that an organisation purchasing a package must determine its functional requirements and weigh these against various products, as shown in the ideal model, this may not happen as expectations are fuelled by a combination of selling acts and organisational learning. Moreover, as a result, expectations of package functionality may also shift when the realities of implementation and usage are experienced.

3.5.2.3 Packaged Software and Organisational Change

It is argued that packaged software is more likely to require a degree of organisational change than custom development (Lassila and Brancheau, 1999). As discussed earlier, this is because packaged software is inscribed with ways of work. Moreover, packaged software has finite capabilities unlike custom approaches and therefore package adoption requires changes in ways of work in line with way the software has been built (Lucas et al., 1988). Furthermore, because packages are embedded with ways of work, organisational changes in line with this imply learning, as discussed earlier (Hanseth and Braa, 1998). This may of course also mean 'unlearning' ways of work and the realisation of this may cause an organisation not to select a package. For example, a major sportswear company the author has worked with wanted to dynamically perform stock re-allocations. In effect, they wanted to reduce certain customers' orders to satisfy those which were perceived as more important. The packages they evaluated would not allow them to do this so they decided not to implement one. Packaged software has further been associated with intended or unintended change[9] (Hanseth and Braa, 1998; Hanseth et al., 2001) and explicit or implicit change agendas (Glass, 1998; Light, 2000) mostly

[9] This despite them being supposedly predictable due to their pre-built standard nature.

in the form of BPR activities. Therefore, in terms of evaluating packaged software, the 'change agent' potential of the software and the various agendas and capabilities for the exploitation of this cannot be ignored.

3.5.3 Customisation Considerations

Packaged software is generally tailored to the specific requirements of an organisation (Klaus et al., 2000). The notion of tailoring needs to be explained as the terminology surrounding the manipulation of packages is diverse. For the purposes of this study, configuration (sometimes called tailoring) and customisation (sometimes called modification, custom fitting and tailoring!) are used. Configuration refers to the 'switching on and off' of functionality that is part of the blueprint of the software (Holland and Light, 1999b). Customisation is used to describe changes or additions to the functionality pre-built into a package (Light, 2001a).

A number of early studies of packaged software indicate that there was an intention to customise products as a short cut to development – and thereby deal with the applications backlog (Weing, 1984; Trauth and Cole, 1992). Indeed, in one study, a whole category of analysis is devoted to problems with product customisation (Gross and Ginzberg, 1984). To clarify this, the subject's problems were that they were unable to customise the software easily, rather than that they had difficulties following the customisation activity. However, from this point, and more strongly in the 2000s, packaged software is linked with the minimisation of customisation activity (Dolmetsch et al., 1998; Brehm et al., 2001).

Even though the decision to implement packaged software carries with it the idea that development is effectively outsourced to the vendor, further development work may be required. Packaged software can be customised to meet organisational requirements. However, the more extensive this becomes it is argued, the more economies of scale are traded away. At some point, heavy customisations become more expensive than custom development (Gremillion, 1982; Lynch, 1984)[10]. Customisation activity may also result in the loss of vendor support for the product or at the very least, a lack of support for the customisations (Golland, 1978; Markus and Tanis, 2000). Moreover, customisations might not work properly after a patch or upgrade (Butler, 1999). Despite the customisation of packaged software being heavily discouraged, many adopters undertake this as a result of a lack of knowledge about the package or cannot avoid it even though they may experience difficulties in getting the result to work (Brehm et al., 2001; Light, 2001a).

Packaged software adoption costs may therefore rise in the same way as they have in custom development projects (Remenyi et al., 1997). In the case of packages, this might be due to the performance of customisations where they were not necessary because of a perceived lack of functionality or 'incomplete' requirements during the process of selection.

3.5.4 Maintenance and Upgrade Considerations

In order to avoid problems of invalidating packaged software support, an organisation may attempt to minimise customisation and rely upon the vendor to develop the package in an appropriate fashion. However, this means that those in the consumer organisation may have to live with the vendor's decisions about

[10] Assuming that custom development was more expensive in the first place.

maintaining and upgrading their software products (Bansler and Havn, 1994). Usually, because of the desire for vendors to maintain economies of scale and a generic product, they will determine who gets involved in this. As shown in section 2.5, SAP chose customers that suited their product development objectives for their R/3 3.1 and 4.0 releases, rather than aiming to satisfy the requirements of their user base.

It is also necessary to point out that maintenance can be a form of revenue generation for the vendor. For example, patches may be provided at no cost, beyond any maintenance agreement. However, changes to smooth out operable, but badly fitting or designed functionality often form the basis of new releases for which additional charges are made (Sawyer, 2001). For example, Microsoft will release service patches for its windows products via the web at no charge. However, changes aimed at improving interfaces and that bring new features are generally (although not always) released as a new version requiring customers to pay an upgrade fee.

Therefore, the maintenance and development trajectories of vendors are once again highlighted, as they were in section 2.6, to be an important consideration when evaluating a package. Furthermore, this discussion makes the agenda of the vendor more explicit than the ideal model and supports the argument for a view of selection that recognises multiple realities and other drivers of the process in addition to that of the consumer organisation.

3.6 Packaged Software Selection: The Final Decision

It has been suggested that presenting a strong business case will attract management support and user participation in the selection process (Kunda and Brooks, 2000). However, this implies that information systems staff are the driver of the selection process and may even be the ones who make the decision. Certainly, information systems staff may be involved in the selection process (assuming they are present in an organisation), but they may not necessarily make the final decision. Many studies related to packaged software suggest that the primary decision makers in this environment tend to be non-information systems senior managers (Sawyer, 2001). At Siemens Power Corporation for instance, the information systems staff worked on collecting the necessary information on which top management could base a sound decision for a particular package (Hirt and Swanson, 1999) and at 'NIBCO' the selection of package was made by eight, predominantly director level, non-information systems members of staff (Brown and Vessey, 2001).

So, the literature reports the presence of two decision making groups – information systems staff and non information systems senior managers, with the latter appearing to carry more weight. Moreover, as amply illustrated earlier, the factors driving the purchase decision are much more complicated than the 'ideal' models would suggest. The decision will be made based upon complex and sometimes highly personal criteria. Furthermore, the criteria might not be agreed with, or perceived as the most sensible by other people involved in, or affected by, the decision. Again, this is something long understood in relation to custom development (Wilson and Howcroft, 2000).

3.7 Packaged Software Implementation

There is a noticeable absence of literature relating to the success and failure of 'pure' packaged software implementation. However, as mentioned earlier, it is possible to examine the literature related to ERP packages and indeed much work on success and failure has been undertaken in this respect. Consequently, the discussion that follows is very much rooted in the ERP literature. However great care has been taken to exclude factors that are specific to the ERP application type and include factors that are appropriate to packaged software in general. A number of studies that offer frameworks for understanding the process of implementation of ERP packages are shown in Table 3.4.

In summary, these essentially begin with some form of needs analysis (comparable with the selection processes described in section 3.2). This is followed by other implementation activities such as the software configuration, business process change and the roll out of the application. A bedding in process follows as those in the organisation become accustomed to the new package and then it begins to be exploited as necessary. Moreover, there is a deluge of literature that aims to facilitate packaged software implementation, specifically in identifying factors which may contribute to success and failure. In the following sections, I discuss the contributions of these studies in relation to the general factors that can be drawn from them. However, as this study is concerned with the process of selection, the specific focus is upon how the factors are linked with this. Note that, at this stage, I make no attempt to critique the nature of the links made. I merely wish to highlight the conventional wisdom reported in the literature. I shall return to these issues in chapter 6, section 6.6.

Table 3.4: Typical Process Models of ERP Packaged Software Implementation

(Esteves and Pastor, 1999)			
Selection	Implementation		Use and Maintenance
(Parr and Shanks, 2000)			
Project planning and software selection.	Identification of modules for implementation through to installation.		Repair, extension and transformation.
(Ross and Vitale, 2000)			
Design	Implementation	Stabilisation	Continuous improvement and transformation.
(Markus and Tanis, 2000)			
Build case for purchase and selection of the software.	Configuration of the software and roll out to the organisation.	Transition from 'go live' to 'normal conditions'	Capture business benefits (if any) from the ERP system and plan the next steps for technology implementation.

3.7.1 User Involvement and Acceptance

The need for user involvement is probably the most widely reported factor that contributes to the success of packaged software implementation. This is seen as important in the belief that those who will interact with the package must have some input in order to determine functionality requirements and facilitate change processes (Gremillion, 1982; Markus and Tanis, 2000; Al-Mudimigh et al., 2001; Akkermans and van Helden, 2002). User/client acceptance is also included here. This is because if users are not involved in the process of selection, requirements may not be determined properly leading to inadequate product selection. There may also be resistance to change and lack of system acceptance (Gibson et al., 1999; Holland et al., 1999a; Holland and Light, 1999b).

3.7.2 Appropriate Business Process Change

This factor involves determining and performing the necessary changes in ways of work to conform with the standard processes embedded within a given package in order to maximise the reported benefits over custom development (Lucas et al., 1988; Holland and Light, 1999b; Markus and Tanis, 2000). As this is often a key goal, it would therefore be expected that the selection process would take into account this factor, and that those involved would aim to choose a package that most closely resembles their intended ways of working (whether this is currently similar to the existing ways of working is another matter).

3.7.3 Top Management Commitment

This factor is generally used to make that point that engaging senior management keeps the project moving, it may also involve the development of a good case for implementation (Avital and Vandenbosch, 2000; Al-Mudimigh et al., 2001; Akkermans and van Helden, 2002). In terms of packaged software selection, it has been demonstrated earlier that senior managers are most likely to make purchase decisions and it would therefore make sense for them to be enrolled in the process. Moreover, even if selection and implementation are seen as separate processes, implementation may prove difficult if 'top management' have not been involved in selecting the package.

3.7.4 Personnel Capabilities

Packaged software will require the recruitment, training and retention of staff to implement and use the package (Holland and Light, 1999b; Markus et al., 2000; Skok and Legge, 2001; Kraemmergaard and Rose, 2002). At the stage of selection,

this may involve internal and external staff in requirements gathering and product evaluation activities. The assumption here is that those involved in the selection process (whether users or external consultants) can assist in choosing the 'right' package only if they are of the right 'calibre'. Moreover, as demonstrated earlier the package selected may have implications in respect of the availability of the personnel capabilities required to implement, operate and maintain the package.

3.7.5 Understanding of the Capabilities of the Package

Understanding the capabilities of the package is highlighted as being important throughout the process of implementation (Lucas et al., 1988; Markus and Tanis, 2000; Al-Mudimigh et al., 2001; Akkermans and van Helden, 2002). The implications for the selection process are that the package needs to be fully understood in order that the most suitable product is purchased. If the most suitable product is purchased, the belief is that this will facilitate implementation as acceptance is increased.

3.7.6 Appropriate Decisions Regarding Customisation and/or Configuration

Packaged software is usually selected on the basis that customisations will not be performed. However, sometimes customisation may be unavoidable (Lucas et al., 1988; McKeen et al., 2002; Holland and Light, 2003). Many packages also require configuration. Where this occurs, it is argued that implementation plans use configuration strategies to facilitate implementation. Plans may include the roll out of partial functionality, rather than full functionality, in 'difficult' implementation conditions or where a faster, simpler, implementation is desired (Holland and Light, 2003). The link between selection and implementation here is that, through the

selection activities, products may be assessed in terms of their need for customisation and/or the potential for implementing numerous configuration strategies.

3.7.7 Recognition of Legacy Information Systems

Even though packages are often implemented to standardise and attain best practices, the literature suggests that explicit recognition of the legacy information systems can facilitate implementation as this is the context for any new development(Lucas et al., 1988; Al-Mudimigh et al., 2001; Holland and Light, 2003). In terms of selection, knowing the 'as is' situation (part of requirements gathering) may assist those in organisations to appreciate the extent of change that may be brought about through implementation.

3.7.8 Sound assessment of User Requirements

Knowing the requirements for the package for implementation purposes is argued to be necessary in order to scope the project (Lucas et al., 1988; Gibson et al., 1999; Markus and Tanis, 2000). Clearly, changes in requirements during implementation could increase timescales and costs. A sound assessment of needs is, therefore, highlighted as important at the time of package selection.

3.7.9 Management of Change and Expectations

Finally, the implementation of packaged software is associated with varying degrees of change and it is argued that the expectations of organisational members need managing in this environment (Markus and Tanis, 2000; Akkermans and van

Helden, 2002). The selection process therefore links with implementation in this fashion via notions of user involvement and moderating the various sales pitches that might be made in order to obtain the package/package sale.

3.7.10 Summary

In summary, it is clear that, however implementation is characterised (either to include selection or treat it as a separate process), there are implications for those activities that arise from selection activities. Therefore, theories of packaged software selection need to take this into account but at present those that emulate the ideal model do not explicitly do so. As reported elsewhere:

> "Our overall experience however suggests that far from being a "take it off the shelf, plug it in" experience, the installation of packaged software has hidden costs that only become apparent during implementation" (Lynch, 1984: 234).

Many of these 'hidden costs' can be traced back to the selection process, and before.

3.8 The Conceptual Framework

Eisenhardt (1989) suggests that theory can be used as an initial guide to data collection. In order to begin the empirical work, it was necessary to develop a range of theoretical bases to help guide the study. The conceptual framework shown in Figure 3.2 is based upon the review of the literature in this and the preceding chapter. Indeed, it has been suggested elsewhere that literature reviews represent the foundation for research and that they facilitate theory development (Hart, 1998; Webster and Watson, 2002). However, any theory could obscure other viewpoints. As Silverman (2000) notes, theories provide the impetus for research by provoking enquiries about the unknown, but they can be self-confirming in that they instruct us to look at phenomena in particular ways. Therefore, the principles of the

hermeneutic circle and dialogical reasoning were used to question and modify initial assumptions and theories (as discussed in much greater detail in chapters 4 and 7). In line with Walsham (1995b), the initial conceptual framework presented here was open to be expansion, revision or disposal.

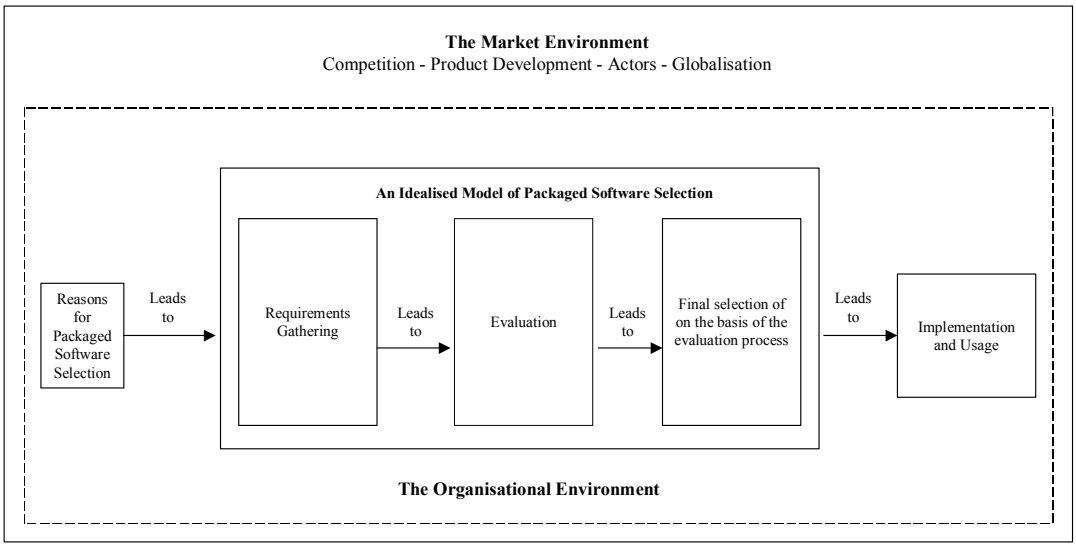


Figure 3.2: The Conceptual Framework

At the heart of the framework is the ideal model of packaged software selection as drawn from the literature. Although packaged software selection theory tends to adopt a rationalistic view, other aspects of the literature suggest otherwise. The wider literature on packaged software, and its selection has therefore been drawn upon to begin to create a richer theoretical picture of packaged software selection that can be used to guide the rest of the study.

The framework first raises to the surface the reasons for packaged software adoption as described in section 3.2. It further suggests how these might 'trigger' and influence the selection process. Similarly, the idea that the process of selection has

implications for implementation and usage as discussed in sections 3.5.2-3.6 is incorporated. These issues have been grouped into the category of the 'organisational environment'. Within the realms of the conceptual framework, the organisational environment is situated in a 'market environment' – the subject of chapter 2. The framework therefore raises the idea that packaged software selection processes are subject to direct and indirect market influences clearer. These influences include those associated with competition in the packaged software industry, product development, the various actors (customers, consumers, vendors and their intermediaries) and globalisation.

The framework simplifies a very complicated area and it could be viewed as rationalistic. However, at this stage, it is merely a guide for the research and a stimulus for the interpretation of the multiple realities of packaged software selection. It should not be taken to be the way to conduct the packaged software selection process.

3.9 Summary and Conclusions

This chapter began by discussing the reasons why packaged software may, or may not, be purchased. These reasons where shown to be highly subjective and in some cases far from considered. In summary, the discussion demonstrates how experiences of packaged software, prior to the enactment of the 'formal' selection process, can be shaped by a range of factors. Following this, the chapter considered the process of packaged software selection as reported in the literature resulting in an ideal model of selection. This model is unpacked, and the processes of requirements gathering, package evaluation and the final selection decision are shown to add further complexity. The chapter then considers how the rest of the process of

implementation can be impacted by selection decisions. These discussions further highlight the problems of rationalistic views of packaged software selection processes. Moreover, they provide support for the development of an alternative, and arguably more realistic, theory which recognises the uncertainties and multiple realities of the process.

The next chapter introduces the research process for the fieldwork and theory development in the thesis.

4 The Research Process

4.1 Introduction

This chapter is concerned with the approach to this research. It could have been divided into two – one focussing upon the paradigms and the other, methodology. However, the intention is to demonstrate the inextricable link between the two and how they can be confused.

The chapter begins by introducing the research tradition in the information systems community. This will be followed by a critique of the underpinnings and application of positivism and interpretivism. Proceeding this, will be an explanation and analysis of the research approach for this study which is to adopt a combination of action research and case study research. The approach to data collection, theory development and an role of the research are then discussed. Finally, there is a discussion of the evaluation of the research.

4.2 The Research Tradition in Information Systems

There is still little agreement on 'the' paradigm for the information systems community resulting in the formation of a number of communities of meaning. These include positivism, post-positivism, critical theory, interpretivism and constructivism. Within information systems research, the two most popular are positivism and interpretivism, often characterised as objective/subjective, qualitative/quantitative, outsider/insider and etic/emic, see (Lee, 1991). However, critical perspectives common in Sociology (Travers, 2001), have begun to receive more attention in information systems, see (Adam, 2002; Brooke, 2002). For the purposes of this study, the focus will be on the differences between interpretivism and positivism but specifically, the interpretive approach of this study.

In the field of information systems, the positivistic paradigm and its associated methods continue to be the most widespread (Goles and Hirschheim, 2000). In 1991 one study suggested that 96.8 per cent of the 155 articles surveyed, were underpinned by positivism (Orlikowski and Baroudi, 1991). However, this picture is changing. By 1995 it was reported that various groups of interpretive researchers had emerged to make up an 'interpretive school' although it was noted then that positivism still dominated and it remained to be seen if interpretivism would take off (Walsham, 1995a). Recently, it has been suggested that information systems has reached the stage where many different research methods and approaches (e.g., quantitative or qualitative, positivist or interpretive) are accepted (Myers, 1999). In further support, the introduction to a special issue of Management Information Systems Quarterly, one of the leading information systems journals reads:

> "An unstated objective for the Special Issue was to publish excellent intensive research simply to increase the legitimacy of such research. From its inception until quite recently, the academic information systems field has often been hostile to non-quantitative and non-positivist research. We found, however, that this goal had already been accomplished well before publication of this first instalment. Indeed, the acceptance of intensive research has been so total that serious challenges to the legitimacy of such research no longer arise. Today, intensive researchers have substantial representation on the editorial boards of major information systems journals."(Markus and Lee, 1999: 35).

The development of information systems in this fashion bodes well for interpretive research. Yet, despite the reporting of its uptake and pockets of high profile research, positivism still remains the most dominant paradigm in information systems. It has even been suggested that positivism is still viewed by many as the 'normal science' (Truex III, 2001). A large contributor to this, is the dominance of the US as a major segment of the information systems research community. Moreover, it has been suggested that this may well have been fuelled by information systems researchers gravitating towards business schools (which are argued to have a tendency to advocate the paradigm of positivism) and because the backgrounds of many also included computer science and engineering (Goles and Hirschheim, 2000).

In addition to arguments of paradigm, a further issue that occupies the information systems research community is that of relevance. The question here is does information systems research have to be relevant?

Briefly, some researchers such as Davenport and Markus (1999) argue that information systems research needs to become more practitioner oriented but this does not mean a resulting lack in rigour. Mitroff (1985) also states that good

research should operate with high academic and high 'street' or organisational knowledge and that this requires mutual respect. I concur with the idea but actually think the relevance debate is a red herring. This is because the question of relevance to whom needs to be asked. Is research supposed to be relevant to society, those in organisations, the information systems community or the researcher? Thus, it is possible that all research is relevant. The main argument that appears in information systems is that research needs to be relevant outside of the research community. In response, there may well be research that does not appear directly relevant to those in work organisations. However, the 'supply chain' of ideas from highly theoretical and into practice arguably makes them indirectly relevant. In any case, this research is very relevant. The topic is arguably of interest to the information systems research community and practitioners. Moreover, the use of action research has been suggested to be directly relevant as it implies learning on the part of practitioners and researchers (Truex III, 2001).

4.3 Paradigms of Inquiry

A paradigm may be viewed as a set of basic 'beliefs' that represent a worldview that defines for its holder, the nature of the world, an individual's place in it and the possible relationships to that world and its parts (Guba and Lincoln, 1994). Burrell and Morgan (1979) further use the term to mean a commonality of perspective which binds the work of a group of theorists together. Moreover, it has been argued that the fundamental beliefs of inquiry paradigms can be summarised by the responses given to three inextricably linked questions - answering one constrains how the others may be answered. These questions are shown in Table 4.1.

Table 4.1: Fundamental Questions for Determining Inquiry Paradigms

1. The ontological question. What is the form and nature of reality and, therefore, what is there that can be known about it? For example, if a "real" world is assumed, then what can be known about it is "how things really are" and "how things really work." Then only those questions that relate to matters of "real" existence and "real" action are admissible; other questions, such as those concerning matters of aesthetic or moral significance, fall outside the realm of legitimate scientific inquiry.

2. The epistemological question. What is the nature of the relationship between the knower or would-be knower and what can be known? The answer that can be given to this question is constrained by the answer already given to the ontological question; that is, not just any relationship can now be postulated. So if, for example, a "real" reality is assumed, then the posture of the knower must be one of objective detachment or value freedom in order to be able to discover "how things really are" and "how things really work."

3. The methodological question. How can the enquirer (would-be knower) go about finding out whatever he or she believes can be known? Again, the answer that can be given to this question is constrained by answers already given to the first two questions; that is, not just any methodology is appropriate. For example, a "real" reality pursued by an "objective" inquirer mandates control of possible confounding factors, whether the methods are qualitative (say, observational) or quantitative (say, analysis of covariance).

(Guba and Lincoln, 1994: 108).

Issues of ontology, epistemology and methodology can therefore be seen as facilitating the exposition of the characteristics of researchers from a paradigmatic perspective which sets the boundaries for what is deemed to be legitimate inquiry. The remainder of this section is devoted to developing a critical understanding of positivism and interpretivism.

4.3.1 Positivism

Positivism is fundamentally concerned with the view that true knowledge is scientific, in the sense of describing the coexistence and succession of observable phenomena (Bullock et al., 1988). This infers that the positivistic researcher is in a position from which they are able to speak for the phenomena observed (Truex III, 2001). Proponents of positivism also assert that scientific knowledge is only obtainable from data that can directly experienced and verified between independent observers (Susman and Evered, 1978). Furthermore, this view propounds that

research, in fields such as information systems, should emulate how it is done in the natural sciences (Lee, 1999b). Positivism assumes that:

- "Objective" reality can be captured
- The observer can be separated from what is observed
- Observations and generalizations are free from situational and temporal constraints; that is, they are universally generalizable
- Causality is linear, and there are no causes without effects and no effects without causes
- Inquiry is value-free (Denzin, 2001: 44).

The aim of positivistic inquiry is explanation that ultimately enables the prediction and control of phenomena, whether physical or human (Guba and Lincoln, 1994). The positivist approach therefore involves the manipulation of theoretical propositions using the rules of formal logic and the rules of hypothetico-deductive logic, so that the theoretical propositions satisfy the four requirements of falsifiability, logical consistency, relative explanatory power, and survival (Lee, 1999a). As this study has a qualitative bent it is perhaps worth noting here how positivistic assumptions would influence this. Qualitative researchers who hold positivistic views are said to favour building techniques into their studies which emulate those of natural scientists or quantitative researchers. For example, they may aim to count the instances they observed, in order to make claims about the representativeness of their findings and also favour collecting large amounts of data (Travers, 2001). Many positivists have, therefore, argued for and against the ideas and practicalities of theory building from qualitative case study data based upon the

argued need for large numbers of cases (Yin and Heald, 1975; McClintock et al., 1979; Yin, 1981; Eisenhardt, 1989; Eisenhardt, 1991; Yin, 1994).

Despite its success in the natural sciences, it has been suggested that the positivistic paradigm is not the only, or indeed always the most appropriate, basis for information systems research (Galliers and Land, 1987). They argue that the factors that can be studied under laboratory conditions are limited and that there are difficulties in reproducing a 'real world' environment. They suggest that a study of decision-making aids on the decision-making behaviour of a manager can only be properly studied in the real world decision-making environment where, for instance, it may be noisy and stressful. Moreover, they propound that the requirement to give variables values, leads to the elimination of factors that are difficult to value. However, these factors may be the ones with the most value. The problems of attempts by positivistic researchers to control and manipulate variables as in the laboratory conditions of the natural sciences have been further critiqued in terms of epistemology. It has for instance, been argued that:

> "the transformations in organizational life through computing are so multifarious as to encompass the most disparate cause-effect relations in different contexts. There is no reason why computing should not result in deskilling in some settings and the enhancement of job content elsewhere, or in greater responsiveness to public diminished responsiveness in others. Indeed, one might well expect quite different effects to ensue from what appear to be the "same" causes in similar or even identical organizations, according to contextual changes in such things as the environments in which organizations act. In short, we see no reason to believe that any simple set of theoretical relationships can account for all the data that one might expect empirical inquiry to bring to light on these subjects."(Attewell and Rule, 1984: 1190)

Here, Attewell and Rule question the hypothetico-deductive logic embodied in many positivistic studies. Smith (1989) further illuminates the problems of positivism by

arguing that attempts of researchers at attaining, and proclaiming objectivity are illusory and damaging to research outcomes. He argues that positivism ignores the inevitable act of interpretation by the researcher and that because of the need for objectivity (and therefore the inevitable attempts of the researcher to distance themselves from the phenomenon) the researcher is not sufficiently close to the phenomenon under investigation to begin to understand it.

4.3.2 Interpretivism

In contrast to positivism, the interpretive position considers the methods of natural science to be inappropriate where human beings are concerned, mainly because different people will interpret the same situation differently (Braa and Vidgen, 1999). Research of an interpretive nature adopts the position that our knowledge of human action is a social construction and therefore an objective reality cannot be discovered by researchers and then replicated by others (Walsham, 1993). Walsham continues, stating that theories concerning reality are ways of making sense of the world and that shared meanings that occur are a form of intersubjectivity rather than objectivity. Interpretivism is therefore concerned with approaches to the understanding of reality that assert that all such knowledge is a social construction and thus subjective (Schwandt, 1994). Moreover, the phenomenological tradition of subjectivity argues that a world constructed by an individual subjectivity is as 'real' and as 'true' to that individual as any reality or truth imposed from outside (Bullock et al., 1988). However, care has to be taken to avoid forms of interpretivism that emphasise nihilism. Indeed, Myers argues that interpretivists may assume that an independent reality exists and therefore they can also be a realist (Myers, 1997). The difference, in contrast to positivists, is that interpretivists assume that access to

reality is only through social constructions such as language, consciousness and shared meanings, which are not neutral.

Interpretivism offers a good basis for conducting relevant research because it can facilitate the development of understandings of the realities of organisational life. If organisational life is treated in the same way as a laboratory experiment, with the aim of the reductionism of problematic situations. The problem is that views become obscured. Thus, much of what can be termed 'relevant messes of interpretive origin' that people encounter on a daily basis may be ignored. The latter point is emphasised below:

> "In social life there is only interpretation. That is, everyday life revolves around persons' interpreting and making judgments about their own behaviors and experiences and those of others. Many times these interpretations and judgments are based on faulty or incorrect understandings. Individuals, for instance, mistake their own experiences for the experiences of others….. But often the understandings that these programs are based upon bear little relationship to the meanings, interpretations, and experiences of the persons they are intended to serve. As a consequence, there are gaps or failures in understanding." (Denzin, 2001: 3).

However, interpretivism is not without critique (Fay, 1987; Gibbons, 1987; Schwandt, 1994). Schwandt (1994) suggests that there are problems of criteria. He asks, what is an 'adequate warrant' for a subjectively mediated account of intersubjective meaning? This is concerned with how the decision is made regarding which is the most interesting and plausible account? He infers that interpretive work could be seen as holding little value as accounts become subject to charges of solipsism (these are only my accounts) and/or relativism (all accounts are equally good/bad, worthy/unworthy).

It is further argued that interpretive accounts lack critical interest or the ability to critique the very accounts they produce. The basis of this is that, historically, interpretive researchers have been directed to 'discipline' their subjectivity and only go 'marginally native'. The problem with this is that such distance means that the research will find it difficult to undertake any critical evaluation of the social reality it seeks to portray (Fay, 1987; Schwandt, 1994).

Also, defining interpretation as an act of inscription vests authority and control in the inscriber (the researcher) and suppresses the dialogic dimension of constructing interpretations of human action. This refers to the idea that, as the researcher is usually the one that 'writes up' the research, it will, inevitably be their interpretation that is put forward (Schwandt, 1994).

Interpretivism leaves no room for an examination of the conditions which give rise to the actions, rules and beliefs which it seeks to explain. It will usually be desired to study these factors as well as the resultant meanings (Fay, 1987).

It has also been stated that interpretivism neglects to explain the pattern of unintended consequences of actions. By definition, these cannot be explained by referring to the intentions of the individuals concerned (Fay, 1987).

4.3.3 A Comparison of Positivistic and Interpretive Paradigms

The questions of ontology, epistemology and methodology as outlined in section 4.3 are useful for further understanding the differences between the positivistic and interpretive paradigms.

The positivistic ontological position is that reality exists independently of our construction of it (Walsham, 1995b). This implies that reality is objective and, as was implied earlier, the researcher is independent of any area of investigation. In contrast, interpretivism adopts the position that the knowledge of human action is a social construction, and therefore an objective reality cannot be discovered by researchers and replicated by others (Walsham, 1993). As suggested in section 4.3, this leads to very different epistemological positions which are summarised in Table 4.2. Therefore, the theory developed from this interpretive field study for example, should be viewed as a means for negotiating a shared understanding of the process of packaged software selection and not the process for packaged software selection.

Table 4.2: Differences Between Positivist and Interpretivist Epistemology
From: (Myers, 1997: 243).

Epistemological assumptions of positivism	**Epistemological assumptions of interpretivism**
1. Experience is taken to be objective, testable, and independent of theoretical explanation	1. Data are not detachable from theory, for what counts as data are determined in the light of some theoretical interpretation, and facts themselves have to be reconstructed in the light of interpretation
2. Theories are held to be artificial constructions or models, yielding explanation in the sense of a logic of hypothetico-deduction (if T is true, phenomenon x follows)	2. In the human sciences theories are mimetic reconstruction of the facts themselves, and the criterion of a good theory is understanding of meanings and intentions rather than deductive explanation
3. Law-like relations (generalisations) are derived from experience and are independent of the investigator, his/her methods and the object of study	3. The generalisations derived from experience are dependent upon the researcher, his/her methods and the interactions with the subjects of study. The validity of the generalisations does not depend upon statistical inference 'but on the plausibility and cogency of the logical reasoning used in describing the results from the cases, and in drawing conclusions from them' (Walsham, 1993, p.15).
4. The language of science can be exact, formalisable and literal	4. The languages of the human sciences are irreducibly equivocal (because of multiple, emergent meanings) and continually adapt themselves to changing circumstances
5. Meanings are separate from facts	5. Meanings in the human sciences are what constitute the facts, for data consist of documents, intentional behaviour, social rules, human artefacts, etc., and these are inseparable from their meanings for agents

Finally, it was mentioned earlier that methodological choice is inextricably linked to ontology and epistemology. However, it needs to be made clear that 'method' is not necessarily linked. Indeed it has been argued elsewhere that methods of data collection and analysis (survey, experiment, case study, etc) are, to some extent at least, independent of the underlying epistemology (Myers, 1997). However, in the information systems community, links between paradigm and method can be confused:

> "By the "interpretive approach," this paper refers to such procedures as those associated with hermeneutics, phenomenology, and case studies. By the "positivist approach," this paper refers to such procedures as those associated with inferential statistics, hypothesis testing, mathematical analysis, and experimental and quasi-experimental design." (Lee, 1991: 342).

To return to the point made earlier, it is necessary to focus upon the link between methodology and paradigm because it is the way that methods are used, and sometimes combined, that gives an inkling of the paradigmatic assumptions of the inquirer. With this in mind, although the methodology that follows next is 'typically' interpretive, it is acknowledged that a similar methodology could have been constructed from a positivistic position (but it would have had different ontological and epistemological emphases). Thus, the next section begins by explaining the consistency between the construction of the methodology and the Interpretivist views on ontology and epistemology.

4.4 The Methodology

The methods used in the conduct of this research are qualitative, case study and action research. However, before this matter is explored in more detail, it is perhaps prudent to raise a point of clarification. As mentioned in the last section, interpretive research is often equated with qualitative approaches and this connection is probably the most widespread. However, this combination of philosophy and methodological choice is not the only possibility. This is a popular misconception in information systems research. There are examples of interpretive research that use quantitative methods (Kaplan and Duchon, 1988; Orlikowski and Baroudi, 1991) and positivistic research based upon qualitative methods (Holland et al., 1994; Wastell et al., 2001). There is further confusion about the use of qualitative and quantitative data and this

is discussed in section 4.4.3. The choice of the case study and then the action research method described in this chapter is therefore one that was deliberate. This choice was heavily influenced by the nature of the investigation as it emerged.

It is possible to claim, because of the use of case and action research, that this research answers the call for methodological pluralism (Gable, 1994; Mingers, 2001). Yet, this would be disingenuous. Even though multiple methods have been used in studies before, such as action research combined with organisational ethnography (Lau, 1999), this was not my initial intention. When the study began, the aim was to undertake case study research. Additionally, there was no intention to use the case of T.Co. Indeed, the study began some time before there was any contact with the company. Moreover, the study was originally concerned with a particular packaged software application type, CRM software (Light, 2001b) – despite guidance that suggests an essential starting point for any research is deciding on and clarifying the subject for study (Mumford, 2001). As Braa and Vidgen (1999) assert, in-context information systems research, regardless of tradition, has unintended consequences, and a degree of uncertainty in respect of how the research project progresses. As the study proceeded it became clear that there was utility in focussing upon packaged software rather than an application type (although interesting in its own right).

Fortunately, in June 2000, the opportunity to work with T.Co arose. The Managing Director was interested in a funded collaborative project that fell within the Teaching Company Scheme. Without telling the story contained in the next chapter, the Managing Director wanted to improve the company's approach to the use of

information systems. By November 2000, a TCS programme had been awarded. The project involved three staff of the Information Systems Research Institute at the University of Salford offering 'technology transfer' opportunities to T.Co. over two years. Early in the project, there was talk of using packages but there was also a strong history of the usage of custom development and it appeared this would continue. In research terms the study began using participant observation, but very quickly a shift to action research was required. This crystallised when the decision to implement packages was made and I was asked for specific input due to my research profile in the area. Contact with the company for the purposes of this study ended in December 2002 and hence 2 years and 6 months have been spent 'in the field'.

Clearly, the methodology, the description of the methods used in the research (Bullock et al., 1988) was, as a result, emergent. Moreover, it can be viewed as bricolage with myself as researcher, the bricoleur. A bricoleur is a kind of professional do-it-yourself person (Lévi-Strauss, 1966). The bricoleur produces a bricolage, that is, a pieced-together, close-knit set of practices that provide solutions to a problem in a concrete situation. However, the job that is completed may not be the one initially undertaken and can therefore be viewed as an emergent construction (Weinstein and Weinstein, 1991). As Nelson et al. (1992) describe, it is pragmatic and self-reflexive. They argue that if new tools have to be invented, or pieced together, then the researcher will do this. Furthermore, the choice of which research practices to employ depends very much upon context and is not set in advance. However, it has been argued that bricolage can be read as a sign of methodological

and theoretical impurity - a symptom of pathology that must be diagnosed, explained and somehow remedied (Schwandt, 2001).

The process of research is dirty, how could this be argued against in the light of the thesis? Yet, it is unhelpful to imply that it is a disease that needs curing. Those that make criticisms of this kind are in denial of the realities of research - at least in information systems and other disciplines involving the study of social situations. They run the risk of focussing too much upon methodology at the expense of contribution. As Pettigrew notes:

> "Contrary to the way the practice of research is often taught and written up, the activity of research is clearly a social process, not merely a rationally contrived act. Further, is it more easily characterized in the language of muddling through, incrementalism, and political process than as a rational, foresightful, goal-directed activity" (Pettigrew, 1985: 222).

It is necessary to remember that the purpose of methodology is to allow others to see how conclusions have been reached, it is not an end in itself. Methodology is no substitute for intellect (Paul, 2002).

This study began with an intrinsic stance in that the aim was to develop a better understanding of the cases in question. However, this rapidly became an instrumental case study whereby the cases became of secondary interest and merely played a supportive role in facilitating an understanding of packaged software selection processes (Stake, 1994). As data collection progresses and new opportunities for research arise beyond the study reported here, there is little doubt that shifts back and forth between intrinsic and instrumental positions will occur.

In this study, the instrumental perspective has no doubt been directly fuelled by the shift towards action research. The field study and consultancy modes of research, outlined by Stowell et al. (1997), clearly require a degree of instrumentalism. Both require, to some extent, a focus upon issues over a better understanding of the case in question. For instance, with the field study mode, it is argued that very rarely will an organisation allow time for a researcher to take up time of employees and other stakeholders with no good reason. Furthermore, with the consultancy mode, a particular problem is the focus and the researcher will be 'lucky' if they learn something from this.

This section now expands upon the detail of the methodology through a discussion of the theory of case study, action research and an overview of the data collection methods, theory development processes and the role of the researcher.

4.4.1 Case Study Research

A brief review of the literature highlights confusion regarding what case study research constitutes. Case research has been categorised variously as qualitative research, ethnographic research and field work, for instance. However, it is clear that the case study is not a data collection technique nor a philosophical stance. Case research can be qualitative and quantitative in nature, it can employ a variety of data collection techniques such as interviews, focus groups and questionnaires. Case study research ma also require the access of secondary data sources such as company reports and statistics. As a form of research, case study is defined by interest in individual cases, not by the methods of inquiry used (Stake, 1994).

One definition of case study is:

> "..an empirical enquiry that: investigates a contemporary phenomenon within its real-life context: when the boundaries between the phenomenon and context are not clearly evident; and in which multiple sources of evidence are used." (Yin, 1989: 23).

Although Yin's work is located in the positivistic paradigm the wording above is inclusive enough to incorporate interpretive case study work. Interestingly his later definition becomes less inclusive incorporating positivistic terminology such as triangulation (Yin, 1994). As Gummesson (1991) points out, the aim of case studies is not a superficial establishment of correlation, but to reach a fundamental understanding of structure and process. Case research is generally concerned with questions of the how and why kind (Yin, 1994). It describes real-life contexts where an intervention has occurred and where there will not be a single set of outcomes.

This leads to the types of case studies that can be conducted for research purposes: exploratory, descriptive and explanatory (Gummesson, 1991). Exploratory case studies are pilot studies that contain data that can be used to formulate more focussed research questions or find out more about an emerging area of interest. Descriptive case studies aim to describe what happens in a given context in order to further understanding of the area under investigation. Explanatory case studies offer different explanations for the same set of events with the aim of indicating how these may apply to other situations. In interpretive terms – the data from the cases are used to offer generalisations in terms of rich insight - broad and diffuse implications (description) and the generation of theory (explanations based upon shared meanings) (Walsham, 1995b). This study has also led to the (further) development

of a concept in the same way as ‘informate’ arose from Zuboff’s work (Zuboff, 1987).

Most criticisms of case research originate from those that believe in the assumptions of positivism and these centre on upon four issues - construct validity, internal validity, reliability and replicability and external validity (Numagami, 1998). Construct validity is concerned with how well the argument being made is supported by the evidence in the study. Internal validity is the measure of how plausible the argument is in relation to competing explanations. Reliability and replicability relate to the requirement that the same findings may be observed repeatedly if the same research procedures are followed. External validity is concerned with the extent to which the findings of a study are generalizable to a wider population (Hedrick et al., 1993; Graziano and Raulin, 1997).

These criticisms become apparent in statements made by those critical and supportive of this type of research. Yin (1989) states that case research has come under attack due to the lack of rigour associated with it, such as sloppy investigator behaviour - including the acceptance of equivocal evidence, and the introduction of bias. MacNealy (1997) draws on several critiques of case research and adds that the method is often misunderstood and so is subject to poor design and enactment. She also concurs with Yin (1989) in that many feel that the case study method offers a less rigorous method compared with those that are based upon 'hard' science. She states that cases are often conducted by one person leading to the potential for bias and also that the results are not generalizable. Miles (1979) expresses concerns with the rigour of case research suggesting that internal and external validity measures are

often developed on an intuitive basis or are simply not well developed. It has been further argued that:

> "we continue to research cases much as we always have - flying by the seat of our pants with little systematic method to guide us." (Stoecker, 1991: 88-89).

Yet, it has to be remembered that this is an interpretive study. Thus, whilst some of these criticisms need careful consideration, others, such as those stemming from the positivistic paradigm, have no place as evaluation criteria. Indeed, the case method has a lot to offer. Advocates of case research argue that it enables the researcher to develop grounded theory that is relevant and practically useful (Eisenhardt, 1989), it is useful for observations over a period of time (Miles and Huberman, 1994), and it allows an investigator to retain the meaningful characteristics of real life events (Yin, 1989).

4.4.2 Action Research

The birth of action research is attributed to Kurt Lewin and the Tavistock Institute (see (Rapoport, 1970)). In the information systems community it has gained ground over the past twenty years with the dominant, though not exclusive, philosophical stance of interpretivism in evidence (Baskerville and Wood-Harper, 1996; Lau, 1999). Action research combines the generation of theory with changing the social system through the researcher acting on or in the social system. The act is presented as the means of both changing the system and generating critical knowledge about it (Susman and Evered, 1978). The aim is to gain learning and knowledge through making deliberate interventions in order to achieve some desirable change in the organisational setting (Braa and Vidgen, 1999).

Further more action research:

> "stems from the behavioural sciences based on the principle that the researcher is within the field of that research and becomes a partner in the action and process of change." (Wood-Harper, 1985: 178)
>
> and:
>
> "aims to contribute both to the practical concerns of people in an n and to the goals of social science by joint collaboration within a mutually acceptable ethical framework." (Rapoport, 1970: 499).

These views clearly distinguish action research from case study research. With action research 'deliberate interventions are made to achieve change' whereas with case study research, even though participant observation techniques may be employed, the interventions are not intended to achieve change. Moreover, interviewing and observing people in organisations without intervention is not action research and has been argued to be more readily described as case study research which frequently reports what practitioners say rather than what practitioners actually do (Avison et al., 1999)[11].

To the aims of contributing to the practical concerns of people and to the goals of social science, Susman and Evered (1978) add a third aim - to develop the self-help competencies of people facing problems. This further distances case study research from action research although people may develop 'self help' competencies as a by-product. The interesting point about the addition by Susman and Evered is that this opens up action research as having the potential for those of the critical paradigm. Although, this work was not critical, sometimes there was the desire for, and possibility of, helping employees at T.Co. change the situation in their favour.

[11] What people say they do and what they really do is discussed further in section 4.4.3.

Further guidance is provided by Baskerville (1999) who states that action research located in information systems:

- Aims at an increased understanding of an immediate social situation, with emphasis on the complex and multivariate nature of this social setting in the information systems domain.
- Simultaneously assists in practical problem solving and expands scientific knowledge. This goal extends into two important process characteristics: First, there are highly interpretive assumptions being made about observation; second, the researcher intervenes in the problem setting.
- Is performed collaboratively and enhances the competencies of the respective actors. A process of participatory observation is implied by this goal. Enhanced competencies (an inevitable result of collaboration) is relative to the previous competencies and the degree to which this is a goal, and its balance between the actors, will depend upon the setting.
- Is primarily applicable for the understanding of change processes in social systems.

While action research appears to be an attractive option for information systems it has been suggested that there are practical difficulties of carrying out the research. Rapoport (1970) reports three dilemmas.

Ethical dilemmas revolve around whether a client is acceptable to the researcher. It is suggested that action research should neither allow the researcher to become captive of one organisation nor disregard the interest of specific clients. This is also part of a broader issue of the place of ethics within the study. For example, this study has been anonymised, an embargo will be placed on the thesis to protect the

company and the company was made completely aware that research was being carried out.

Goal dilemmas consider the needs of researcher and suggest that practice needs may dominate. In terms of this research, the needs of T.Co have dominated. Fortunately, these needs have provided excellent insights in the area of investigation.

Dilemmas of initiative suggest that the service oriented nature of action research places the client in the driving seat - they have a problem and need help with it. This contrasts with the ethos of the academy where protections exist to take the immediate ebb and flow of practical pressures off the scholar so that they can conduct the research with minimal interference. There are two considerations here – first, academic life has moved on as there are now many practical pressures that can interfere (Rapoport was writing in 1970!), and second, the initiative very much depends upon the mode of action research being undertaken – consultancy or field mode (Stowell et al., 1997). In field study mode, the primary focus for the researcher is 'the research' with assistance to the organisation secondary. Conversely in consultancy mode the focus is upon improving an organisational situation with the learning from that situation being secondary. The issues associated with each of these modes are shown in Table 4.3Table 4.3, were considered throughout the process of the research.

The rigour of action research has also been subject to further critique because of the (too?) close connections with organisational activity. It has also been suggested that action research may be viewed as little more than consultancy (McKay and Marshall,

2001). Moreover, some may argue that, even where successful, causal connections and explanations cannot be safely made and that researchers may be questioned over a perceived lack of impartiality and a supposed lack of scientific rigour leading to data validity concerns (Baskerville and Wood-Harper, 1996). Consequently, the application of principles for the conduct and evaluation of the research have been defined and applied (see section 4.5)

4.4.3 Data Collection and Reporting

The emergence of case study and action research as the two methods of the study allows for efficiency in the usage of data collection techniques They fundamentally, do not differ too much because participant observation was part of the case study. However, before the approach to data collection and reporting is discussed, it is important to clarify the difference between qualitative and quantitative data. It is argued that quantitative data takes the form of numbers or measurements and qualitative data the form of words (Punch, 2000). To expand, qualitative data are argued to incorporate the use of words to describe situations, individuals or circumstances surrounding a phenomenon, whilst quantitative data uses numbers in the form of counts or measurements to give precision to observations (Remenyi and Williams, 1996). However, the idea that numbers are only used for measurement purposes carries distinctly positivistic overtones, and presents numbers and measures as objective with no meaning. Furthermore, qualitative studies may also involve other data such as observations of actions in order to facilitate understandings of socially constructed situations. Qualitative data are therefore perhaps more usefully thought of as being concerned with *meaning* and quantitative data with that of

measurement. This is why this study includes only qualitative data, even though numbers, such as software costs, may be used in analysis and for theory building.

Table 4.3: Issues for Action Research in Field Study and Consultancy Modes

	Issues for Field Study Mode	Issues for Consultancy Mode
Choosing the domain	It needs to be appropriate to the research. Contact is likely to be initiated by researcher. There may be difficulty in recruiting collaborators. The domain needs to offer sufficient depth and complexity in order that insights can be obtained. Researcher recognition of mutual benefit and shared interest between themselves and the domain if any.	The domain may choose you. Are the clients needs going to be met, is this suitable for research.
Planning and implementing the study	Focus upon and document the approach towards the study rather than predicting and testing outcomes Choose methods appropriate to the epistemological basis of the research.	Client demands may make it difficult to follow the direction the researcher wishes. Develop potential strategies for action prior to the event to limit damage.
Engaging collaborators	Make them aware of the researcher's intentions at the outset. That is to say the researcher's prime interest is in learning about a particular aspect of research from which they, the collaborators, may benefit. The value of the research is dependent upon collaboration – researchers need to consider changes in personnel.	They may be overly interested and you have to fit with how they want the work to progress which may detract from your work.
The rigour of action research:	Record data systematically over time. Knowing what to record and what not to is a problem for the action researcher.	Research based upon consultancy activities may run into difficulties for example if there is no evidence for the assertions made about the work other than the consultants remembrance of it; the cycles of learning cannot be identified because at the time the action research process was not clearly conceived.
False expectations	Researchers need to manage expectations.	The client may not be happy with the identification of the problem by the researcher as it emerges from the process of learning (need to let the client know this).
Identifying useful lessons and bringing the study to a close	It is difficult to identify lessons as actions arise. Documenting actions facilitates the development of lessons. The researcher recognises the point at which closure is reached.	Clients may decide this. What about your research?

It is suggested that in an interpretive field study, primary or secondary sources of data may be used. Primary data may be collected by methods such as participant observation, interviews, tape recordings of meetings, unpublished written materials created independently of the researcher (such as memos, correspondence, minutes of meetings and reports). Secondary source material includes previously published documents such as those by scholars or practitioners (Myers, 1997). In this study, a wide range of data collection methods were employed and sources referred to including:

- interviews which were mostly unstructured with a few being semi-structured,
- participant and non-participant observation,
- opportunistic meetings/conversations,
- minutes of meetings,
- reports/requirements documentation such as that prepared in-house, by consultants and by packaged software vendors,
- email correspondence,
- Company newsletters.

The reader may find it surprising that tape recording was not used. This was not undertaken for two reasons. It was felt that tape recording would interfere too much with the job at hand. At times, I was acting in a consultancy role as well as that of researcher. Furthermore, tape recording might have hindered data collection due to the subjects' perceptions of, and comfort with, tape recording. Based on over two years of working with those in the organisation, this was the right strategy – the Managing Director in particular would never have agreed! Tape recording is no

guarantee of getting more accurate data anyway. Even when peoples' activities are tape-recorded and transcribed, the reliability of the interpretation of transcripts may be gravely weakened by a failure to record apparently trivial pauses and overlaps (Silverman, 2000). Thus, as I had several years of interviewing experience, and good access to the company, it was felt that better insights would be obtained by recording data by hand where appropriate, or salient points directly after 'meetings' where even note taking would have been problematic. This approach also allowed for time to be spent observing what people did as well as finding out about what they said they did. As Silverman states – decontextualized accounts of meanings are very limited guides to the complexities of human action (Silverman, 1998)[12]. Watching, as a form of 'recording', was invaluable in this study.

It is also suggested that reporting on the collection of field data should include details of the research sites chosen, the reasons for this choice, the number of people who were interviewed, and what hierarchical or professional positions they occupied (Walsham, 1995b). To that end the beginning of this section details how and why the site was chosen – quite by accident is the flippant response, but obviously it would not of been chosen had T.Co. not decided to implement packaged software and had there not been a good chance of access in order to the undertake study[13]. The people who were interviewed and their positions are shown in Table 4.4.

From June to November 2000 I visited the company 6 times. From November 2000, when the TCS programme began, I visited most weeks, spending a half to one day

[12] Silverman actually uses the term human-computer interaction, but I think that the assertion retains the flavour post modification. Also, I acknowledge that interviews do offer the chance to observe what people do as well as what they say they do – non-verbal messages for example.
[13] I hasten to add, the decision of T.Co. to implement packaged software way in no way influenced by my desire to study them choosing, implementing and using it.

on site. Documentary evidence and interviews were also used to obtain insights into the position at T.Co. before the study began. These data stretch back to initial thinking regarding the client-tracking project in August 1999.

The data are reported as follows. The story begins by familiarising the reader with the company. Then, a combination of thick and thin description are used to present three information systems projects at T.Co, which involve the selection of packaged software. Thin description states 'facts' and thick description gives the context of an action, aims to brief the reader about the intentions and meanings that organised the action, traces the evolution and development of the action, and presents the action as a text that can then be interpreted (Denzin, 2001)[14]. For example, data about company turnover, organisational structure and a timeline of events are provided, but these are put into context with information about the competitive environment, and peoples' reported feelings about the company and the various projects. It is further suggested that the analysis strategy needs to be outlined (Walsham, 1995b). This is undertaken in section 4.4.4.

4.4.4 The Development of Theory

Eisenhardt (1989) identifies three distinct uses of theory: as an initial guide to design and data collection; as part of an iterative process of data collection and analysis; and as a final product of the research. In this study, theory is present in all three areas.

[14] The process of thick and thin description, and deciding what is thick and what is thin, is an act of interpretation.

Table 4.4: Contact Points at T.Co. 2000-2002

Contact(s)	When Contacted
Managing Director	Throughout the study
Academics x 2 (contributing to the TCS Programme)	Throughout the study
Business Development Managers (BDM) x 2 (one left T.Co during the project)	Throughout the study
Commercial Director	During the Client-tracking and migration projects.
External Consultants x 3	During the Client-tracking and migration projects.
External Software Vendors x 2	During the Client-tracking and migration projects.
Finance Officer Manager	Mostly during the migration project.
Graduate Trainee	Mostly during the client-tracking project.
Human Resources	Mostly during the client-tracking project.
IT Manager (Initially the TCS Associate)	Throughout the study.
Managing Client Consultant	Mostly during the client-tracking project (but some contact for the migration and OneSource Projects).
Marketing Manager	Mostly during the client-tracking project (but some contact for the migration and OneSource Projects).
Non-Executive Director	Throughout the study
Operations Manager x 2 (one left during the project)	Mostly during the client-tracking project (but some contact for the migration and OneSource Projects).
Research Manager	Throughout the study
Sales Manager	Throughout the study
Secretaries x 3	Throughout the study
Several Client Consultants	During the Client-tracking and migration projects.
Sales Supervisor	Mostly during the client-tracking project (but some contact for the migration and OneSource Projects).
Telesales Representatives	Mostly during the client-tracking project (but some contact for the migration and OneSource Projects).

In order to 'get on' with the empirical work, it was necessary, for pragmatic reasons, to develop the conceptual framework (Figure 3.2) to facilitate the research. I could not afford to undertake 'walk in the park research' as such and used theory at this stage in an instrumental fashion. I further added, removed and refined the theory in

response to my increased consumption of the literature and later, in comparison to my empirical work. Indeed, it has been suggested that literature reviews represent the foundation for research and that they facilitate theory development (Hart, 1998; Webster and Watson, 2002). Finally, one of the outputs of the study was a set of concepts with plausible relationships – a theory as defined by Strauss and Corbin (1994). However, the guiding conceptual framework could have obscured over viewpoints. As Silverman (2000) states, theories provide the impetus for research by provoking enquiries about the unknown, but they can be self-confirming in that they instruct us to look at phenomena in particular ways. Therefore the principles of the hermeneutic circle and dialogical reasoning were used to modify initial assumptions and theories (see section 4.5.1).

Figure 4.1 illustrates the processes by which theory was developed throughout the study when working in field study and consultancy modes. In both cases reflection occurs in respect of action and theory and, is fed into and feeds from, action and theory as the study progresses. Therefore, the research process involves reflection, learning and refinement in the light of new data.

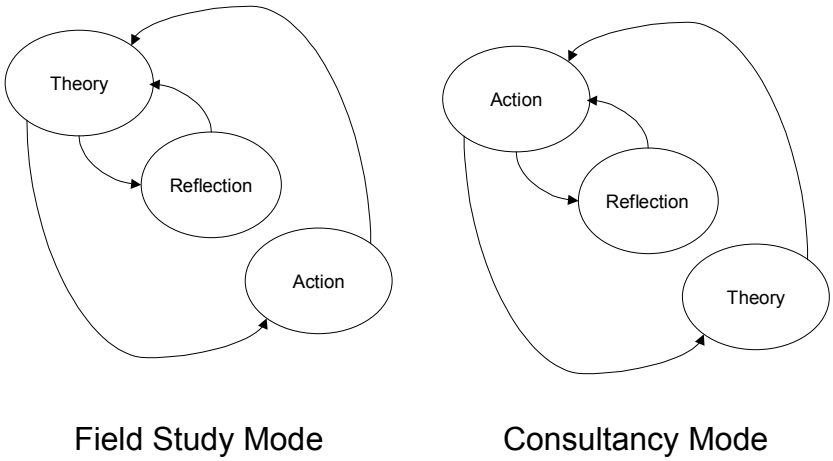


Figure 4.1: Field Study and Consultancy Mode Theory-Action Cycles
(Stowell et al., 1997: 194-196)

In this study, and in contrast to the model put forward by Susman and Evered (1978), action research was acted in field study and consultancy mode in a similar fashion to that reported by McKay and Marshall (2001) who argue that action research may involve two interconnected and interacting cycles. However, even by operating in both modes, a 'best' theory is not the result. In the interpretive tradition, there are no correct and incorrect theories but there are interesting and less interesting ways to view the world (Walsham, 1993).

It was therefore necessary to determine when I had a plausible story to tell that would be viewed as an interesting way to view the world. Thus, throughout the process of theorising whilst undertaking the research, occurrences of theoretical shifts from the ideal model of package selection began to emerge. For example, with the migration project it became clear very early on that a recommended vendor was

going to be the one that was chosen no matter what – this small finding immediately raised questions about the plausibility of the ideal model. As more of these findings emerged, the theory was revised until it was felt that nothing further was being added and theoretical saturation had been reached (Eisenhardt, 1989; Strauss and Corbin, 1994). The iterations of theory development are more fully explained in section 7.3.7 and illustrative examples are provided in Appendix B.

4.4.5 Role of the researcher

(Walsham, 1995b) identifies two roles for the researcher – that of outside observer and involved researcher. From the interpretive perspective, neither of these can be viewed as objective because of the subjectivity involved in the interpretation of interpretations. That is, I was always representing my view of other peoples' views. However, there were varying degrees of engagement experienced throughout the study, as mentioned in the data collection section of this chapter. Sometimes participation was undertaken, sometimes just observation.

Forms of engagement has been conceptualised by Nandhakumar and Jones (1997). Their work suggests a classification of differing forms of engagement as shown in Figure 4.2. The paper suggests that although this classification is useful, the distinctions are may not be so clear cut. For example, given the right conditions a researcher could be more 'engaged' in an interview than via consultancy work. However, they identify two broad groups of methods – those that are distant, such as analysis of published statistics, textual analysis and postal surveys through which there is no direct interaction between the researcher and the research phenomena or where interactions are strictly controlled. At the other end are 'engaged' methods,

such as consultancy, action research and participant observation where the researcher is personally involved in the activities in the research context. They suggest that interpretive information systems researchers might favour the use of engaged data-gathering methods but their study indicates a paucity in this respect.

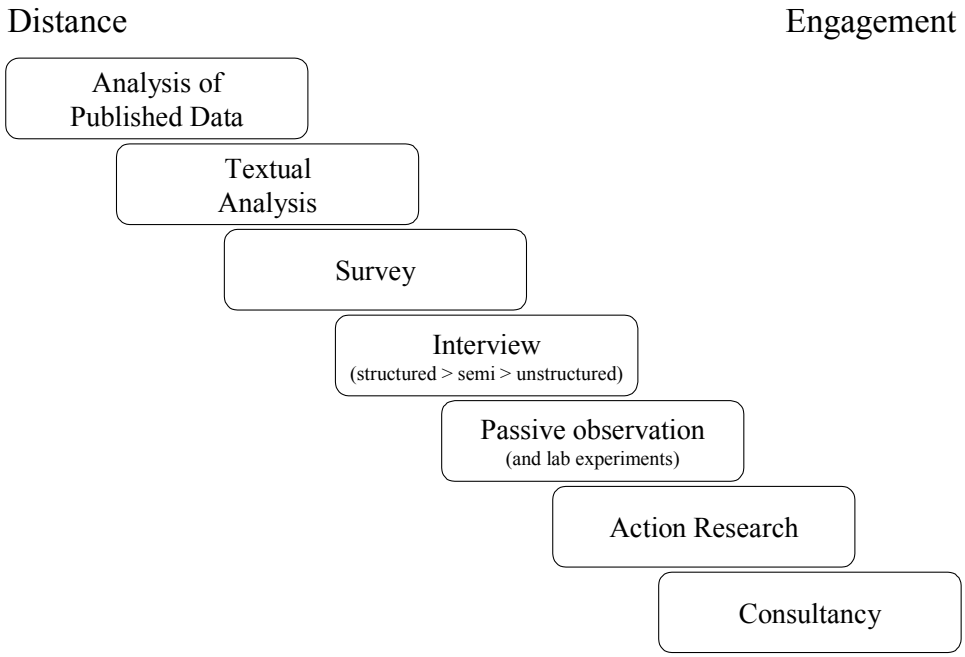


Figure 4.2: Forms of Engagement in Research
(Nandhakumar and Jones, 1997: 113).

In this study, various forms of engagement are evident, from the analysis of published data right through to consultancy led action research. This, multi-engagement approach that has emerged assisted in creating a fairly rich interpretation of the experiences of those at T.Co. It is also important to point out that throughout the study, some form of engagement directly with those of T.Co was in place, even if it was not directly related to this research. The role of the 'outside observer' was therefore really never fulfilled.

4.5 Principles for the Conduct and Evaluation of the Research

For qualitative research, and especially that with an interpretive perspective, the need for evaluation is acute due to the criticisms that are in evidence throughout the information systems research community. This section introduces the principles used to guide the conduct, and perform the evaluation of the study. It begins with a description of the major part of the framework. This is the set of principles for conducting and evaluating interpretive research offered by Klein and Myers (1999). These are then added to as some key principles do not appear to be explicitly covered within the Klein and Myers framework. These are contribution to knowledge and methodological choices. The principles are described here and then used to evaluate the study in section 7.3 of the concluding chapter. These principles were also used to guide the progress of the study as I felt it was sensible to aim to conduct the research in cognisance of the evaluation criteria especially as they contained implicit guidance for the conduct of research.

4.5.1 Principles for the Conduct and Evaluation of Interpretive Field Research

The following principles as per Klein and Myers (1999: 72)[15] have been used to guide the study and will be used in its evaluation:

- **The Fundamental Principle of the Hermeneutic Circle -** This principle suggests that human understanding is achieved by iterating between considering the interdependent meaning of parts and the whole that they form. This principle of human understanding is fundamental to the other principles.

[15] Note that the principle of suspicion is not used as Klein and Myers suggest this is only appropriate to research conducted from a critical perspective.

- **The Principle of Contextualisation -** Requires critical reflection of the social and historical background of the research setting, so that the intended audience can see how the current situation under investigation emerged.

- **The Principle of Interaction Between the Researchers and the Subjects -** Requires critical reflection on how the research materials (or "data") were socially constructed through the interaction between the researchers and participants.

- **The Principle of Abstraction and Generalisation -** Requires relating the idiographic details revealed by the data interpretation through the application of principles one and two to theoretical, general concepts that describe the nature of human understanding and social action.

- **The Principle of Dialogical Reasoning -** Requires sensitivity to possible contradictions between the theoretical preconceptions guiding the research design and actual findings ("the story which the data tell") with subsequent cycles of revision.

- **The Principle of Multiple Interpretations -** Requires sensitivity to possible differences in interpretations among the participants as are typically expressed in multiple narratives or stories of the same sequence of events under study. Similar to multiple witness accounts even if all tell it as they saw it.

4.5.2 Contribution to Knowledge

Doing research implies that a contribution to existing knowledge should be made (Paul, 2002). Hedrick et al. (1993) state that good research typically requires substantial up front work, which provides an understanding of the problem or issues. Not only does this provide a focus for the research, it also enables the selection of appropriate topics that have not already been investigated and it may facilitate the convergence of activities and interests (Webster and Watson, 2002). Moreover, Campbell et al. (1982) state that good research is likely to be borne out of intuition rather than any logical analysis. Hence, although the literature was used initially to assist in scoping the study, as the research progressed, what were perceived as possible opportunities for obtaining interesting insights, were taken when they arose. More specifically, Myers (1997) argues that a good indicator of interpretive research is that it offers a contribution to the information systems community in respect of whether the author has developed new concepts, applied new concepts or applied a well known theory in a new or unique way. He also suggests that a good marker is research that offers rich insights into the human, social and organisational aspects of information systems development and application or if it contradicts conventional wisdom. Gill and Johnson (1991) also note that good projects will display symmetry of outcomes. That is, whatever the result it will be useful. The research was therefore, conducted with these points in mind.

4.5.3 Appropriate Methodological Choices

Lawler (1985) argues that research must be approached using a method relevant to the area under investigation. At the heart of determining, an appropriate methodology therefore is the required outcome. Does the research project aim to be

inductive or deductive? Inductive studies aim to draw general conclusions from empirical observations whilst deductive studies are concerned with generating conclusions through a process of logical reasoning (Ghauri et al., 1995). Therefore, does the research project want to produce theories or hypothesis (induction) or accept and reject hypotheses (deduction)? An error in methodology is going to lead to the wrong type of outcome. For example, a quantitative survey relying upon hypethetico-deductive methods (Miller, 1991; Sayer, 1992) would be inappropriate for a researcher wanting to generate frames of reference as as with theory building from case studies (Eisenhardt, 1989; Strauss and Corbin, 1994). Although it is necessary to carefully consider which method or methods are appropriate for the area under investigation, caution should be exercised to ensure that this is not the focus of the research effort. It has already been noted that an over zealous attitude in this area can adversely affect research outcomes. Moreover, Buchanan et al. (1988) propose that a preoccupation with theoretical and epistemological aspects of research can be disabling and may result in research not even beginning. The methodology should be a facilitator of the research process. Several authors state that rarely do the rhetoric of research methods reflect the reality (Pettigrew, 1985; Paul, 2002).

4.6 Summary and Conclusion

Despite the fact that the social impacts of computing have been described as infinitely variable (Attewell and Rule, 1984), positivism, that aims to control and manipulate variables, is the dominant paradigm in information systems research. However, this chapter notes that other approaches such as critical perspectives and interpretivism that recognise and welcome infinite variability, are gaining ground in mainstream information systems research. This interpretive field study uses case

study and action research methods to generate rich insight and theory regarding issues associated with packaged software selection. In order that the insights and theory were allowed to emerge, the methodology itself was emergent and has been characterised as a form of bricolage. This has involved numerous data collection techniques with accompanying degrees of formality and engagement. The next chapter presents the story of T.Co. and the proceeding one offers an analysis of the situation in terms of rich insight and theoretical contribution.

5 Packaged Software Selection at T.Co

5.1 Introduction

This chapter presents data from a field study of the selection, implementation and usage of a number of packaged software products at a Consultancy company – T.Co. The story is situated within a major information systems project, which includes a migration from an Apple Mac environment to a Windows environment and a complete overhaul of application support for the business. The initial projected budget was £50,000 – the cost to date (4th Quarter 2002, including projected Goldmine implementation February 2003) is around £184,500 (this does not incorporate staff time, merely the costs associated with the software, hardware and external consulting). Added to this, is a further combined cost of £77,500 per year to cover the various subscription fees, running, maintenance and support of the new infrastructure. The chapter begins with an introduction to the company. It then offers an interpretation of three concurrent and interrelated projects – the migration of applications from an Apple Mac to a Microsoft Windows operating environment, the purchase of OneSource for the Research Department and the purchase of

Goldmine to support client-tracking activities throughout the enterprise. This chapter provides the basis for the next chapter, which offers an analysis of the package software selection experiences of T.Co.

5.2 An Overview of T.Co

T.Co is a UK based consultancy company that was formed in 1990 by the Managing Director. The headquarters for the company is in Manchester, and in 1999, it has opened a second office in Reading. In the June 2000, when this study began, the turnover of the company was £1.1 million. At this time the company employed 19 internal staff and a further 20 external consultants who aided service provision. In the run up to 2000 the company was enjoying a rapidly increasing turnover and had plans for further growth. New offices were scheduled to be opened in Leeds, Bristol and the West Midlands. The role of T.Co is to provide clients with the skills to improve their performance in their current position and/or assist in the conduct of job-search campaigns. These service offerings are discussed in detail in the next section.

5.2.1 Service Offerings

Originally, T.Co provided a service for senior managers who wanted to improve their expertise or change career direction. The initial offering was a mail order curriculum vitae (CV) and accompanying letter preparation service. However, by the end of the first year of business, there was a recession in the UK economy and as a result, a new market for T.Co services emerged. Historically, T.Co had directed its offering at private clients. With the onset of recession, the company found that blue-chip companies were approaching them and wanting to purchase their services as

part of redundancy packages for their employees. That is, a company might want to make a manager's position redundant, and therefore to show they were a good employer, they would pay T.Co. (or another provider) to assist their employee in finding a new job. Corporate customers are now the major source of T.Co business. Furthermore, as the company has evolved, so have the services that include:

- Individual outplacement - working with individuals to find them employment;
- Individual performance improvement - working with individuals to improve their management capabilities;
- Group outplacement - working with groups to find them employment;
- Group performance improvement - working with groups to improve their team working capabilities.

These services comprise components such as interview skills training, team building, change management counselling, communication training and personalised research provision. The components are variously packaged. The Executive program is a six month long one-to-one programme were a consultant works closely with a client to improve their chances of finding suitable employment. The Senior Executive programme comprises the same support but for an unlimited period and is usually targeted at senior clients. The Director programme extends the Senior Executive programme, is tailored to the client and project managed by the Chief Executive. A more recent addition is the 'Rapid' programme that captures the middle management sector where clients are provided with three month's guidance and access to a multimedia job search product. Finally, Group Outplacement targets blue-collar workers. This involves the set up of an administrative and job search resource centre

at the sponsoring organisation's premises. Those clients using the performance improvement service are segmented in a similar fashion, and tap into the similar service components. Most of T.Co's business is in the areas of individual job search and performance improvement.

5.2.2 Competing in the Environment

In 2000, the UK market for outplacement services was valued at £80 million and T.Co had a very small share of this on a national basis. However, in the North they enjoyed a 10 per cent share of that market. The Managing Director stated that by 2005 the target was for a 5 per cent share of the UK market. The company's two main competitors held 30 and 25 per cent market shares respectively. In 2000, both had greater national coverage than T.Co, although by the end of 2002, this gap was reducing.

T.Co's strategic aim is to improve productivity and profitability whilst maintaining the high quality of service that their customers expect. The market place is extremely competitive and in order to operate effectively in this environment, T.Co focuses upon personalised service, especially in respect of their search services. For example, tailored client CV mailings are sent out to prospective employers. The Research Manager believes that the 'personal touch' and the 'can-do' attitude that all clients receive throughout their time at T.Co is a major contributor to their widely reported 100 per cent success rate for clients. Speed of client placement into work is a further component of the differentiation strategy. As the Managing Director makes clear in a newsletter:

> "The work our Research Department does for our clients is skilled, labour intensive and expensive. But the investment is justified. We know it gives our clients a real advantage – the chance to be in contact with perhaps hundreds of potential named employers very early in their job search. It reduces the hard slog, generates interviews and allows much more time for personal networking and information gathering. And, above all, it shortens job search times."

5.2.3 Organisational Paradigm and Structure

The underlying paradigm of the organisation is strongly unitary. Organisational goals reign supreme, are dictated by management, unconditionally accepted and worked to by employees. There is definitely an ethos of a football team. Managers are frequently heard, stood in the middle of departments, proclaiming: "how do we solve this problem", "come on we have to pull together to reach this months target" and "this is what our clients expect". The company is organised in a hierarchical fashion with strong control and command structures a shown in Figure 5.1.

Indeed, a key feature of the organisation is surveillance. This is argued to be necessary by senior management in terms of service delivery – monitoring client progress for example. Hand in hand with this however, is a desire to ensure that employees do what is necessary to ensure that the monitoring takes place and that there are clients coming through the door to monitor. An extreme example of the surveillance culture in the organisation is the code of conduct for the use of computers at T.Co. This states "the company may monitor all incoming and outgoing emails.. You should not assume that email is private" and "employees should be aware that all internet access is monitored". Additionally, the research log application (discussed later) exists solely to monitor staff performance.

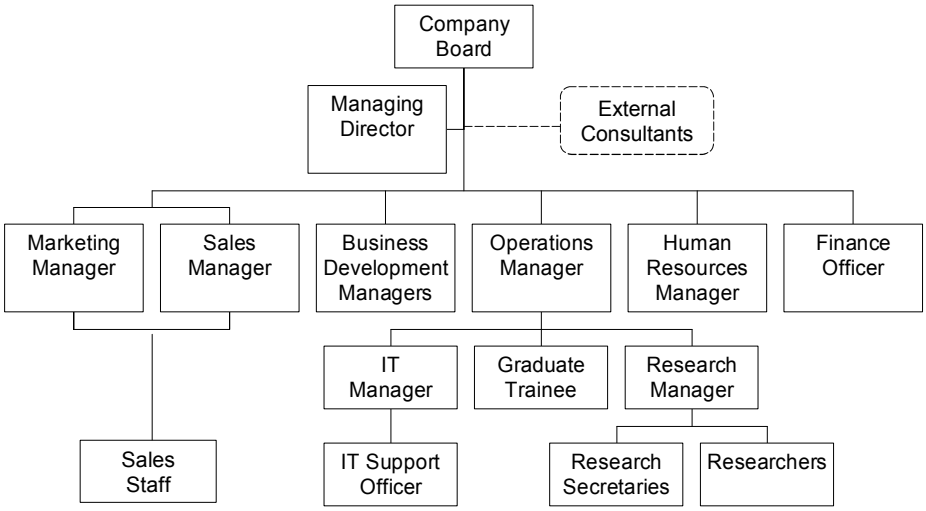


Figure 5.1: The Organisation Structure of T.Co

The organisation is led by a **Board** consisting of a Managing Director who sets and communicates company direction; a Chairman[16] who regulates the direction of T.Co and identifies control and reporting systems; further non-executive Board members offering various skills and expertise, and regional Business Development Managers (BDMs) who ensure that regional offices have input at Board Level.

The **Sales and Marketing** department, in conjunction with the BDMs, are responsible for generating and maintaining business for T.Co. This involveds identifying prospective sponsors and managing the relationship with them should they purchase the services of T.Co.

The **Research Department** assists clients in finding out about prospective employers. They introduce new clients to the range of services available at T.Co

[16] T.Co's terminology rather than Chairperson.

(dependent upon the product they are subscribed to). They also undertake research regarding prospective employers on behalf of clients.

The **External Consultants** mentor the clients in their job search and/or performance improvement activities. They deliver much of the T.Co product. Consultants are self employed and some even work for T.Co competitors.

5.2.4 A Process Map of T.Co

Figure 5.2, illustrates the complexity associated with the ideal company business model. Normally, the process would begin with obtaining market information about the potential sponsors. These sponsors will then be approached and if this is successful, a client (or clients) would appear at T.Co. The client then begins a process of mentoring for performance improvement or job search activities. Throughout this process of mentoring, it is necessary to keep the sponsor informed of progress made with the client, and the outcome of the process. In the ideal model, the outcome is performance improved and employment gained. There is of course the potential for other outcomes. It is also important to note that clients may eventually become sponsors, sponsors may become clients and both may use T.Co's services more than once. This makes it important that the experiences they have with T.Co are good ones. Note also, that this process is complicated further by the need to coordinate activities across functional departments and with external business partners such as the Consultants.

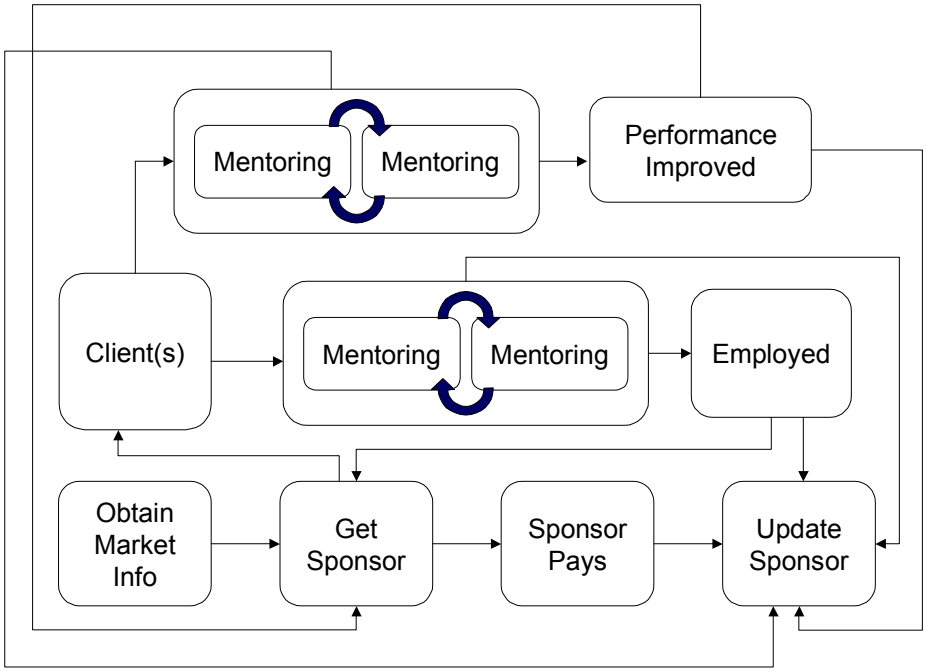


Figure 5.2: A Process Map of T.Co

5.2.5 The information system Infrastructure

An Apple Mac environment was in place in June 2000. The infrastructure consisted of an Apple server supporting twelve Apple PCs at Manchester and three PCs at Reading via an ISDN line. Two further PCs (for client use) and four staff laptops were also in place, but they were not connected to the network. Internet access and email were available via a further ISDN link. The applications in place were Microsoft Office 98 (office automation), Sage 4.1. (Accounting) and a range of custom applications created in the Filemaker Pro 4.1 database environment. These consisted of:

- **Targeted Mailing Programme (TMP) –** This provides data for client research. Primary research entails the client passing over a brief devised from their consultant mentoring meetings to the research department. This brief

incorporates details such as type of role and industry in which the client wishes to obtain employment. The research department searches the TMP application (which has a repository of organisational data, for organisations that fit the brief). These data are collated into a 'headline research report' and passed back to the client. The client amends this further and the secondary research is performed. This entails checking that the most up to date details (such as name of targeted person in employing organisation and their position) are contained in the TMP database. If they are over three months old, these are updated by telephoning the organisation in question on behalf of the client. When these data are complete, the clients' CV(s) and covering letter(s) that have been prepared will be sent out.

- **Senior Search –** This is a home-grown database of recruitment companies and head-hunters. The database is used to identify possible employers for those on the Director and Senior Executive programmes.

- **Research Log –** This maps the research activities from when the research team receives a brief through to the point where contacts are checked. The log is used to create a spreadsheet for monitoring the efficiency of the team. This is then transcribed onto the whiteboard in the Operations Managers office for performance management purposes.

- **New Clients –** This holds the capability to record details about T.Co clients and their progress through the service to which they are subscribed. These details include the client's start date, the service they are subscribed to, how they found

a job, the success rate of the client in their approaches to prospective employers and where they eventually commenced work (much of these data are missing).

- **Prospects Labels –** This contains functions to create labels for various departments.

- **Corporate Order Book –** This is used to track invoicing processes and as the basis of a first contact with Corporate Clients by the sales team.

- **Corporate –** This is the main application used by the Sales Team. All corporate contacts are kept in this repository. The application also contains histories of communications, such as meeting dates and directions to premises.

- **Order Book –** This performs the same functions as the Corporate Order Book, but for private clients.

Usage of the applications, by-department, is shown in Figure 5.3. Obviously, these applications were all operating in the Apple Mac environment. In addition, the Research Manager said "the company had very limited knowledge of information systems". Moreover, I was informed in an interview with the Commercial Director, that she had initially designed and built many of the Filemaker Pro applications. Furthermore, she was quite open about the fact that she did not have an information systems development background. The applications had been further developed by the Research Manager, her previous incumbent and the Finance Officer. All of these staff were self-taught. The applications were flat file and stand alone leading to

duplication and inconsistencies in data entry throughout the organisation. Despite this, the Managing Director had a vision for the company that relied very much upon competent execution of information system support. It was suggested by the IT Manager, that "the company sees the development of product and services as continuous process, aligning supply and demand in a profitable manner and that this had considerable implications for their IT infrastructure."

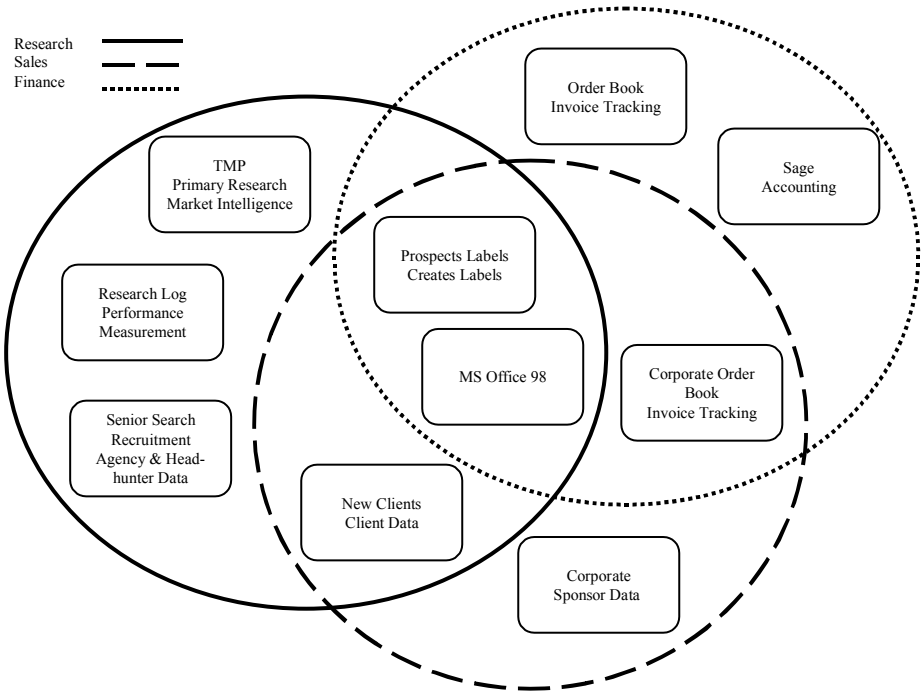


Figure 5.3:Applications by Department

5.2.6 The Evolution of information system and Organisation

When the company was first formed it operated only the mail order CV service. This required word processing software (part of Claris Works) to prepare CVs and covering letters. There was also a basic 'Head Hunting' application (Senior Search). As the company's services at that time were focussed upon individuals, these simple applications supported a very simple business model. Interestingly the company's

first infrastructure was Apple Mac based as the Managing Director had a history of working in the publishing industry where he said Apple Macs had dominated. He claimed they were "better machines". However, this model has evolved dramatically over the past 12 years, as have the information systems.

A BDM informed me that the first phase of the evolution was to add performance improvement services and this required collecting additional data about clients. The Filemaker Pro applications were therefore modified to support sales and marketing activities such as producing mailing packages of company literature. Data about client progress through the service were also collected. T.Co then began to grow quickly and significantly. Along with this grew expectations about information systems in the organisation. In 1996, the question of moving from the Apple Mac and to a Windows environment was raised. According to the IT Manager "This was the biggest investment decision to date that the company had made in respect of IT."

However, the Apple Mac environment remained, based upon external advice. The company was informed by external consultants that Apple Mac cross platform capabilities had come a long way since they had first implemented them and that this would not be an issue. By this time, T.Co had expanded to two sites (Manchester and Reading) and the network was expanded accordingly. The expenditure for the 1996 upgrade was £35,000. The Research Manager reported that the infrastructure initially worked well with the support of a local IT vendor. However, the company predominantly offered support to local schools and their services represented this. The company could not cope with the demands of a growing business such as T.Co. The Research Manager suggested that in reality, T.Co out grew the company almost

immediately with their needs becoming more sophisticated. Furthermore, the Managing Director stated that “we did not realise the importance we would place upon our vendor at the time, quite simply because we didn’t have the knowledge or experience of working with IT to this extent.”

The lack of available organisations that supported Apple Macs also became a limiting factor so T.Co became locked into the local company who were not proactive with future forecasting. The Managing Director said that “T.Co required a partner who had the capability to contribute towards the IT strategic vision of an expanding company and what they actually had was a hardware provider.”

However, Filemaker Pro was still perceived by various members of staff at T.Co as being an ideal product because as their needs evolved, so did the database, through the addition of fields and creation of new layouts. In the requirements gathering process for the client-tracking project (discussed later), staff said: “we like Filemaker Pro because it does what we want” (Research Secretary), “I know how to use this now, it took some time to get used to it, but now it’s fine” (Sales Secretary) and “three clicks and I’ve got what I need, and 'S' is so good at doing the techie stuff I could get anything I wanted from the system” (Research Secretary).

As a side effect, a culture of user led development emerged. Many users were willing and able to offer feedback about the applications and even undertake hands on development tasks such as the creation of the new fields and layouts. The advantages of user led development have also been interpreted as a disadvantage. The IT Manager stated that “Although Filemaker Pro is easy to adapt and minimal

skills are needed to develop the database, database disciplines are not existent and the Filemaker Pro databases are not documented. A realistic comment is that they have run on and on without a driver."

In June 2002, T.Co had several pools of data that were duplicated and dirty. At the same time, managers wanted to undertake more sophisticated analysis of those data however, the IT was not capable of facilitating this. For example, the Sales Manager said "if we are going to expand, I need to have my finger on the pulse of the business!" Instead, analysis involved many manual processes (such as collating hard copy documents and re-keying information) and it was reportedly labour intensive, and not as accurate as it could have been. However, the company did have 11 years worth of data that they could call upon, although these were in various repositories. This was characterised as by a secretary as "a huge security blanket that they were reluctant to let go of."

Indeed, in 1996 when there had been the opportunity to switch from the Apple Mac based Filemaker Pro environment, the pull was so strong that it remained. The IT Manager has since reflected upon the decision at the time and believes they made the right choice at that time. She said that, given what she had learnt about the history of T.Co, the company would not have been capable of dealing with a migration project at that time especially as the move from the Apple Mac and to the Windows environment would have been technically difficult. In 2000 however, she felt that things had changed at T.Co, but more importantly in relation to the technology.

Even in June 2000, however, what has been described by the IT Manager as "the marriage to the Macs" was still evident. Emotions ran high in their defence and comments such as the Macs didn't break down – ever", from long serving members of staff in all departments were commonplace. However, even in light of this, it was clear that the death of the Apple Mac environment was inevitable.

5.2.7 The Motivation for Change

Several issues motivated a need for change in the infrastructure at T.Co. These can be characterised in terms of company growth and capabilities of the infrastructure to support this. In the late 1990s the company defined a strategy that incorporated increased market penetration particularly through wider geographical coverage of that market. Winning more customers in existing areas was also a target according to the Sales and Marketing Managers. These plans brought with them the need to reconsider the capabilities of the current infrastructure. It was clear that things could not continue as they were. For example, Microsoft Word for Apple Mac was in use. This worked until a document was received from someone who had created or amended it in Microsoft Word for Windows. The text would become corrupted and reformat itself, font size would alter and apostrophes would disappear. Consequently, time and effort had to be spent on reformatting the documents. Documents that had crossed platforms also suffered printing problems. There were regular occurrences of documents viewed correctly on-screen, printing incorrectly. In one meeting, the Research Manager highlighted how this had lead to fire fighting:

> "Because of the formatting problems brought about by the Macs, things are always a bit frantic around here. I tend to be most responsive to who shouts loudest at the moment, which is probably not the best way of managing things".

This is the point at which I joined T.Co. At the first meeting to discuss a proposed project between T.Co and myself, the Managing Director asked me if we (he meant T.Co) should move from the Apple Mac environment and to PCs (more accurately – a Windows based environment). He had already told me that there was a big attraction to 'Macs' at T.Co and therefore I thought it prudent to be non-committal at this stage. I therefore mumbled something about pros and cons of Apple Macs, and that I was open minded on the issue. In the same meeting, he asked me how much the 'project' would cost. We had discussed the possibility of a CRM implementation although it was not termed as such at this point. Based upon the data I had about the organisation and a similar recent implementation I had been involved with, I gave him an estimate for the CRM implementation, of £50,000.

Between June and November, time was spent working upon the proposal, getting it accepted and appointing the graduate who would implement it (she is named as the IT Manager in the story). By November however, the decision to migrate from the Apple Mac environment had been taken. The Managing Director had sought the advice of an IT Director he knew and had taken it. He was convinced that the Apple Mac environment was not the standard of the companies they dealt with, and this was the route of many operational problems. Furthermore, the Apple Mac environment and its current support facilities were perceived as not being able to stand the growing business pressures, growing Client and Sponsor expectations, and it was ultimately preventing growth. Problems with the applications had not been identified yet. The platform was the enemy (although not in everybody's eyes as shall be revealed later).

5.3 The Migration Project

The migration project was initially geared toward shifting between an Apple Mac and Windows environment. The breakdown of the software, hardware and services procured is shown in Table 5.1.

Table 5.1: Key Features of the Content of the Migration Project

- Cabling infrastructure enhancement: Expanding the Head-office network to include the new office space, whilst enhancing the network with a flexible and expandable design.
- Windows server core platforms and software: Replacing the existing Apple Mac server with industry standard, corporate class servers and applications for e-Mail, Diary management, File and Print facilities, Intranet and Internet facilities.
- Desktop replacement: Replacing the existing Apple Mac desktop platforms with industry standard small footprint Windows based systems.
- Desktop software upgrade: Changing the appropriate Office software and Database
- Implementation of Virtual Private Network capability.
- Implementation of technical and strategic services support contract.

The total costs for the project were £135,000 with a further bill of £45,000 per year running, maintenance and support costs.

5.3.1 The Migration Project: Finding the Vendor

The migration project team included the IT Manager, the Managing Director, an External Consultant[17], the Research Manager and myself. The main consideration, when sourcing the IT solution, was to find a vendor that was able to suggest the hardware and solve the immediate problem and also a company that would offer advice and that could assist in the strategic evolution of information systems. A good working relationship had to be created so a vendor had to be sourced that could work alongside the team at T.Co. By early December 2000, the requirements of T.Co had been finalised. These were:

[17] Appointed via the IT Director the Managing Director taken the original advice from about the need to migrate.

- A partner who could assist them in upgrading their IT infrastructure, migrate the existing systems and data, and provide on-going support of their hardware and systems.
- Vendors should have a proven record of accomplishment in delivering these services and be capable of providing support to each local office.
- Assistance with the re-cabling of their Manchester office.
- Provision and installation of new PC hardware.
- Installation of a new IT infrastructure based upon Microsoft Windows including: a file and print server, e-mail (local and remote capability), diary management applications, database server, remote network access, secure internet access, virus protection and Wide Area Network links to Reading and Solihull.
- Migration of data from Filemaker Pro applications in the Apple Mac environment to the Windows environment.

The selection of vendor was really a foregone conclusion. The external IT consultant knew a company from a previous role. However, I suggested that the company should look at least at two others for comparability (to make my point I asked the team how many quotes they would get for work they wanted to carry out on their home and this seemed to get the message across). I also suggested these could be used as bargaining tools with the recommended vendor. In the end, two other vendors were invited to tender for the work. The IT Manager told me, that at the vendor presentations, it been agreed by the project team that each response had the potential to offer the final solution but they had strengths and weaknesses[18].

[18] I was unable to attend the meeting.

- Vendor 1, was a large international company that was independently owned, had many different resources available via many different geographical locations. The organisation offered a wide range of products and services and they claimed that 'no project was too small or large' for them to cope with. A mid ranged company,

- Vendor 2, (the recommended provider) had a similar organisational set up as T.Co and was located close to the headquarters. They were also a growing company that offered internal services to remote users. They could support customers remotely but were not large enough to offer the varied levels of support that Vendor 1 could.

- Vendor 3, was sourced from the Internet with no prior recommendations and was described by the IT Manager as the 'budget option'.

The initial contact with each company, after receipt of the tenders, was made by the IT Manager and the external consultant on the vendors' premises. The IT Manager said that it offered a very different insight to how the organisations worked. Vendor 1 was late and it was obvious that no preparation had been made for the meeting. This immediately gave the impression that the project was indeed a trivial one for this large corporation. A sales person was initially involved so answers to technical matters were not available immediately. Moreover, the documentation regarding the project had to be chased, communications were often not answered and the account manager's first contact with T.Co was on the day of selection. The final solution had travelled through three of four different departments before finalisation. Any new

area opened in the project introduced a new department, at a different location and with a different contact. Costs were also high. Vendors 2 and 3 were perceived as offering a more personal approach. Initial meetings included senior management, both companies had a rich knowledge of T.Co and customer service appeared to be a priority for them. A 'beauty parade'[19] was arranged for the three vendors to present their final designs, choice of solutions and meet the Managing Director of T.Co. Although the IT Manager and external consultant were involved in this process, the purchase decision was taken by the Managing Director.

Vendor 1 was reported as presenting in a manner that the team expected. The IT Manager said they offered a generic solution that did not completely address their requirements. There was general agreement that Vendor 3 had presented in a professional manner, answered unpredictable questions from the team and was felt to give an overall good performance. However, the IT Manager and External Consultant had reservations about their abilities in relation to support in cognisance of their site visit where they witnessed disorganised working and storage areas, and what appeared to be inadequate technical support. The IT Manager said "my kids' bedrooms were in better shape!"

Vendor 2 also presented professionally, and critically, the Managing Director from the company attended. According to the IT Manager, mutual respect between him and the Managing Director of T.Co was immediately evident. Their proposal incorporated Virtual Private Network and ADSL technologies and was heralded by the Managing Director of T.Co as "a design for the future!" The recommended vendor was the one that obtained the contract.

[19] A term coined by the Managing Director.

5.3.2 Obtaining Approval

The renewal of technology at T.Co was characterised as a "brave and foresighted decision of the board" by the IT Manager. Clearly, it was going to involve the most significant information systems expenditure to date for the company. However, even though the vendor had been chosen by the Managing Director, the release of the capital still had to be approved by the board. Within T.Co, this required the IT Manager and External Consultant to create a business case for Board approval. In summary, the case highlighted the costs of the Apple Mac environment and how these could be reduced by migrating to a Windows environment. Staff were interviewed by the IT and Research Manager, and it was suggested that around 8.84 per cent of the working week was spent dealing with 'Mac problems' (for any given staff member this might vary from 0 to 22.2% of their weekly workload). A selection of the issues used for justifying the investment is shown in. Table 5.2.

Cost immediately became a major factor as the proposed budget (the figure of £50,000 that had been provided for the CRM implementation) was instantly viewed as overspent by the Managing Director, due to the migration. Even though I had not factored a migration into the figure I had provided originally, this was now taken by the Board to be the amount they were going to spend on the project, including the migration. This resulted in a dialogue between the Board members and the IT Team, which consisted of cycles of cost justification. Throughout this process, a paring down exercise was conducted but eventually, it was the original (more costly) specification that was implemented. This was because equipment, such as servers, that had been ruled out of the plan at first, were reinstated when the Board members realised the business needed them.

Table 5.2: Migration Investment Justification[20]

	Problems	**Anticipated Benefits**
Administration	Documents that are saved in a Windows environment and then opened in an Apple Mac environment often degrade and need to be amended. This may require clients to be contacted meaning that T.Co systems are seen as problematic. The same is true of email traffic. Documents that have crossed platforms may also suffer printing problems. Regular occurrences of documents viewing correctly on screen but printing improperly were cited.	Cross platform glitches will be addressed. Image of T.Co is improved. Staffs working environment is improved.
Research	The use of 'Virtual PC' on Apple Mac computers slows other applications down. Interoperability problems as detailed for Administration. Several applications research requires are only available on PC and therefore the staffs have to use the client PC that can prove problematic.	Fewer interoperability problems will be evidenced. Access to applications will be improved. The migration will lead to a move from flat file to relational database architecture.
Sales	The management of manual diaries is labour intensive. There are nine in total that may need to be arranged in concert. Double bookings occur which is bad for the company image. The ISDN line between Manchester and Theale is not robust leading to problems in communication. Data are data not being backed up onto the server, putting T.Co in a dangerous position.	An electronic diary will make diary management more efficient, effective and accurate. The new infrastructure will improve communications capabilities.
Finance	The Sage package runs on via 'Virtual PC' meaning it is slow.	Sage will operate in a Windows environment much faster.

A further factor (apart from the misuse of the initial quotation) that contributed to an increase in the overall budget for the project (including the client-tracking project) was the rapid growth of the company. Many new staff members, a number of satellite offices and extra remote workers were now present. This had led to a constant reassessment and expansion of requirements in relation to software licences, maintenance agreements, equipment and training. The contact (incorporating costs

[20] Adapted from: Business Case for PC Environment Documentation: February 2001.

and specifications) was eventually signed 28 April 2001 and 'go live' was 29 May 2001.

5.3.3 The Migration of the Filemaker Pro Applications

A major concern was the movement of the Filemaker Pro applications as they supported the key processes at T.Co. There was some discussion amongst the project team about moving the Filemaker Pro applications to an Microsoft Access environment on a temporary basis until the client-tracking project was implemented. The rationale for this was due to the Managing Director's and the Board's desire to ferociously curtail costs at any opportunity. Therefore, as Microsoft Office was being purchased, Access was essentially free of charge, in contrast to Filemaker Pro. Although licences had been purchased for Filemaker Pro in the Apple Mac environment, new ones would need to be purchased for the Windows environment. Furthermore, the project team were aware that eventually, the Filemaker Pro applications were expected to be replaced by the new client-tracking system and therefore it seemed a prime opportunity to cut costs. Moreover, during the migration, it was suggested that the existing applications could have been modified slightly to improve their structure and decrease duplication through the normalisation and merger of several of the applications. This would also have made the future migration of the 'Filemaker Pro' data to the client-tracking system much easier. However, this was discounted by the Managing Director as too risky. He said "there are times to spend and times not to spend! Filemaker Pro is core to our business and it cannot afford to fail. In my opinion, a move to Access would increase our chances of failure.

The Managing Director perceived that a move to Access might make the application stop working and also that it would require more training. He felt that given the major changes the company would experience with the migration, paying for licences that would be discarded later was "a necessary evil"[21].

Therefore, several Filemaker Pro licences for the Windows environment were purchased. Staff were allocated time to test migrated data from the Apple Mac version of Filemaker Pro to the Windows version. A programme of testing was developed for each area, where different forms of data entry and extraction were performed using the migrated data. The IT Manager noted that the testing procedure had offered a double benefit as staff became accustomed to the new system as they tested. Much to the surprise of the team, the migration between the environments was uneventful – the Filemaker Pro database package had versions for Apple Mac and PC and the conversion was simple.

5.3.4 The Shift from Microsoft Office (Apple Mac) to Microsoft Office (Windows)

The migration project offers the first opportunity to consider the packaged software experiences of T.Co in relation to the Microsoft Office conversion. Effectively, there was no selection process. The Office package had been used in the Apple Mac environment and it was perceived as a straight swap. Therefore, no formal requirements gathering or evaluation of potential packages was undertaken. Overall, the process was initially uneventful (especially at the headquarters) and staff welcomed the move to the Microsoft Office in the Windows environment. For many

[21] It has never clear whether the Managing Director really meant this or if he was also 'attached' to the Filemaker Pro environment.

staff, this was a return to familiarity. Most staff that had joined T.Co from other organisations that had used Windows before and it had been necessary for them to learn how to use the Apple Mac environment. There were some staff who had only ever used Apple Macs computers and this was a major change for them. Therefore, a good deal of time was assigned to the task of training. A Non-Executive Director claimed that "No matter what technology was invested in, if people did not know how to use the equipment the project would be a failure."

The IT Manager said that all staff, in all locations and at all levels, would be trained to a common standard by the same trainer. To facilitate this, a training room was built at the company headquarters and sessions were arranged. The IT Manager also reported at the time that not only did the exercise achieve the training objectives but also a dialogue was opened up on the new system. She said that staff from all levels of the company had sat together for a number of days and aired their views and fears (these mostly centred on the move away from the Apple Macs by those that had known nothing else).

5.3.5 Issues Associated with the Microsoft Office Package

There were three key issues associated with the move from Microsoft Office in the Apple Mac environment and to the Windows environment.

The migration involved major reformatting work because of the switch from Word for Apple Mac to Word for Windows. This required all documents to be saved in .doc format (Apple Mac used a different file extension). Furthermore, the opportunity to reformat the corrupted documents was taken at this point. Although

this was a time consuming exercise, staff were willing to engage in this as they could see the time that they would save in the new environment.

Originally, the company only had one external email mailbox. E-mails that came into the company arrived in that mailbox and were then either printed out and handed to the correct individuals or e-mailed internally to the recipient. Moreover, before the migration, the Sales team managed eight paper diaries. One diary co-ordinated in-house resources of room, projector and client PC availability and the remaining seven were used to plan the seven BDMs' appointments throughout the UK. Additionally, the manual transcription of diary entries onto a sheet of paper had to be completed at the end of each day. This was faxed to each BDM every evening in time for use the next day. The diary was also duplicated for those at the various sites of the company. This meant faxing a weekly diary view and telephone conversations as necessary to update the diaries in the light of new appointments, postponements and cancellations. In addition, as each diary was A3 size, the sheer physical movement of these was problematic and they took up valuable office space. Microsoft Outlook was implemented and this (combined with the new infrastructure) gave staff individual external email accounts. The package also offered electronic diary management capabilities. A further package was implemented which allowed Outlook to generate diaries for rooms and equipment availability. The benefits that these applications brought were highly visible to all and they were instantly agreed as a resounding success.

The BDMs were expected to use the electronic diary via the Virtual Private Network. Vendor 2 also accessed their email in this way and informed T.Co that the switch

over would be simple. However, the key difference was, as the IT Manager pointed out, that the users from each company were "worlds apart." The BDMs at T.Co were novices in remote working. Thus, the use of email brought with it a steep learning curve. The BDMs did not have any knowledge about how to make remote working, work for them and others (they had never used Outlook before). For example, despite extensive training, they did not realise that they needed to synchronise their email and other Outlook folders regularly (to stop the activity taking a long time). Another example is related to the Apple Mac environment. Some members of staff had saved Word files in Rich Text Format (RTF), in order to the address problems of compatibility. It was discovered that any document that was saved in RTF and sent to multiple users was quadrupled in size. After a month's usage by the remote users, Outlook came to a to a halt because the BDMs were attempting to download several mega-bites of data at home via a 56k modem connection. Users were also not logging out of the secure connection environment and this meant they were taking up bandwidth even if they were not at their laptops. The BDMs relied heavily upon access to their diaries, were frustrated by the problems they were encountering and blamed the infrastructure. Comments such as those shown below were taken from, and are typical of, those made around the time in meetings and in the offices:

> "this system was supposed to save time, not complicate matters" (BDM 1)
>
> "you just can't trust the technology, this needs sorting and it needs sorting now" (BDM 2)
>
> "I knew this would happen, bring back the Macs!" (BDM 3)

It was not until the IT provider looked into the situation that the problem was identified and remedied. The BDMs never acknowledged they had a hand in the

problem. It was to do with the technology as far as they were concerned. Only seven staff members where involved in this access problem, but they were all senior managers and they were very vocal in their criticisms which fuelled a negative view of the whole migration project for a short time. I heard one of the secretaries, for example, sympathising with one of the BDMs[22] "I know, the whole thing's a fiasco, I much preferred the Macs".

However, for most people there was evidence to suggest that the migration had been perceived more positively. Post migration I asked various members of staff how they felt about the new system and they were generally comfortable with it as the responses below suggest:

> "it's great, no more reformatting of word documents" (Research Secretary)
>
> "it's taken me a while to get used to Windows, but at least I don't have to do all the reformatting like before" (Research Secretary)
>
> "I have my own email account now which is useful" (Supervisor)
>
> "It's good to be working with Windows as I'm learning something that's more common than the Macs were, although I liked them too – especially the I Macs!" (IT Support Officer)

5.4 The OneSource Project

In March 2000, the Managing Director developed a request for proposals. The proposal suggested that T.Co needed assistance in improving their research capabilities in respect of the 'market intelligence' that was obtained for usage by

[22] I was not sure whether she was sympathising because he was a BDM or because she agreed. Unfortunately, as I didn't know her that well, I did not feel able to ask her.

clients in job search campaigns. The rationale at the time was that demands upon the research department were increasing due to:

- An increasing number of clients using its services
- A desire by management to improve the interview 'hit' rate thereby increasing the chance of client success whilst, at the same time, reducing the number of organisations that need to be approached.
- The need to keep abreast of best professional practice in information management.
- The need to keep abreast of latest technological developments, e.g. the Internet.
- Servicing more offices – the forecasted growth was 2 to 5 within the next four years.

T.Co were seeking consultancy advice, guidance in respect of how they could benchmark against best practice, and cost effectively achieve best practice. Specifically, they were interested in the information that would be required to achieve this and asked:

- What information do we need to look for?
- Where do we look for it?
- How do we look for it?
- How do we store it and re-use it?
- How do we transmit it to other offices?

In June 2000, this project had not started. When I began working with T.Co, the Managing Director asked an external consultancy group and myself to tender for the

work. I referred the work to a group of colleagues, but in the end, the Managing Director decided to wait and see what the forthcoming TCS programme would deliver.

5.4.1 The Selection of the OneSource Product

By January 2001, a case had been prepared to justify investment in a product called OneSource. The Research Manager had decided that the department needed a new business intelligence database and, that she really knew what was needed, therefore there was little point in waiting to see the outcomes of the TCS programme. The Research Manager had identified the package, as she had worked with this and other similar packages, in a previous role. OneSource is an online business browser which is linked to a business intelligence database. The database contains over 400,000 UK company profiles such as turnover, staff positions and premises addresses. It also contains news items, business and trade articles, research reports, industrial intelligence and financial data. The primary idea was that the browser would be used to identify prospective companies and contacts that clients could seek work with. OneSource was intended to replace the home-grown equivalent – the TMP application. The TMP application required manual updating by staff at T.Co, data quality was weak and it only held details of around 27,000 organisations. OneSource was trialed by the Research Manager and the positive outcomes of this were fed into the justification document. The Research Manager stated "OneSource offers a competitive advantage in terms of quality of Research and level of service to our clients. OneSource adds a professionalism and efficiency that should turn a Gold Service into a Platinum one."

Her report also recommended the extended usage of OneSource throughout T.Co and the purchase of additional application. These recommendations are shown in Table 5.3.

Table 5.3: Benefits and Recommendations for the Purchase of OneSource

Benefits to T.Co:

- Company information is swiftly collated and can be exported to a number of applications
- OneSource will remove time consuming and labour intensive tasks in the research department
- The application negates the need for further personnel support for the research manager
- The application will be used by others releasing the research manager for value adding work.
- The sales team could use the application to identify potential sponsors.

Recommendations:

- That we subscribe to a ten-licence version of OneSource at £20k pa.
- That it becomes the major source of information for the Research Department
- That all reports are customised as "T.Co Documents"
- That it is installed in the client library at Manchester, Theale and the Midlands.
- One primary researcher is located at each office so that each office does its own standard primary research with secondary research continuing to be carried out at Manchester
- That consultants are trained in its use.
- That the Sales and Telemarketing Department are trained in its use to improve existing standards and intelligence of sales information
- That we replace the T.Co Senior Search database with Executive Grapevine CD-Rom (at a cost of £3,500 per year, plus £600 per satellite office).

The decision to purchase and implement OneSource was agreed with minimal questioning by the Board. Interestingly, I later found out that the Research Manager had shared a two-hour train journey with the Managing Director and it was during this time that she was able to sell the application to him. The TMP application had contained key functionality that OneSource did not. It was therefore decided that, until the client-tracking package[23] was implemented, it would be necessary to retain certain aspects of the TMP application and establish a link between this and OneSource. Essentially, OneSource was to be used as a searchable database. Once searches had been performed to identify potential employers, their details would be

[23] It was expected that the client-tracking package would replace the TMP application in its entirety.

exported to a revised TMP application. The TMP application would then be used to guide secondary research and the generation of mail shots to the employees.

In order to refine the TMP application, and establish the link with One Source, a requirements gathering process was initiated involving a member of the TCS academic team, the IT Manager and the Research Manager. This began in September 2001 with prototyping, testing and refinement following on and carried out up to January 2002. Awareness sessions were then held to obtain feedback from consultants, sales and research staff. This was fed into the process and the new application was released April 2002.

The new application and OneSource both appeared to be received warmly by the staff who used them and they were quickly subsumed into the everyday working lives of those who interacted with them. The Research Manager was particularly delighted as she felt her life was now going to be much easier. She said "I'm so chuffed we got it, no more having to update TMP manually!"

Furthermore, access for staff at satellite offices was appreciated by them, as it had resulted in quicker response times for clients and in searching for potential sponsors. They no longer had to wait for headquarters research staff to undertaken searches on their behalf as the information could be accessed via the internet. For example, one Telesales representative stated "It was a really good idea of 'S''s to recommend that we use OneSource too – it gives us more companies to go at."

The IT Manager was similarly happy, but for different reasons. She said that the purchase of OneSource was a small step in making her life simpler, and it would become more so when the TMP application was also finally replaced with Goldmine (the client-tracking application). She claimed that when it was in place, she could get on with more strategic work and let the day-to-day stuff deal with itself. In a meeting late November 2002, I asked the IT Manager how the application was bedding in and she said it was still fine although she pointed out that staff were forgetting what it was like before. She said the sales staff were now becoming more demanding and were beginning to complain that overseas and public sector organisation data was not available in the database. I asked the same Telesales Representative as before about her feelings on OneSource at this point. She said, "Yes, I'm still happy, although it would be nice to have more global data". Interestingly, they had never had this before, but now it had become a priority.

I did ask if the package was to be rolled out to consultants as initially intended, but I was told by the Research Manager that this had been put on hold as there were only a limited number of licenses available. I therefore asked why more were not purchased and she smiled and told me it just was not an option at the moment. As I suspected, I was told by another source, the real reasoning behind this was that the consultants were not inclined to undertake work that had always been done for them (the searches), even though it might have improved client experiences. It was explained to me that T.Co had decided not to push this as it was more important to keep the consultants engaged in the client-tracking project.

5.5 The Client-tracking Project

For the purposes of this study, the story begins in August 1999 and ends December 2002[24]. It is concerned with the activities surrounding a project that has been variously defined as the job search times project, the client monitoring system, the CRM system and the client-tracking project. However, for simplicity it is easiest to think of it as that concerned with using IT to support the monitoring of client progress at T.Co.

5.5.1 The 'Job Search Times' Project

The client-tracking project at T.Co can be traced back at least as far as August 1999 when senior management at T.Co discussed a 'Job Search Times Project'[25]. In September that year, a meeting was held involving the Managing Director, the Commercial Director and the Operations Manager to discuss this further. As a result of this meeting a requirements gathering process was initiated. The requirements gathering process was conducted by the Operations Manager and the Managing Consultant. They began by mapping out a model client journey by brainstorming the key steps. This initial model is shown below:

- Date of first meeting (between client and consultant)
- Date of approval of CV
- Date of first research brief received (by research department)
- Date of headline research returned to client (by research department)
- Date of headline research approved by client (and given to research department)
- Date of first TMP sent out (by research department)

[24] The project is still continuing and is anticipated to be complete by early in 2003.
[25] Job Search Times Project Memo – September 1999

- Dates of interview clinics (with consultant)
- Date of second TMP sent out (by research department)
- Frequency of meetings between consultants and clients

The model was then validated and refined through discussions with consultants.

Table 5.4 shows the model in February 2000. The idea was to use this to develop a set of requirements for a new 'Client Monitoring' system. The requirements were to highlight client overspends by region and to allow for the development, and implementation of, performance metrics in respect of consultant performance. That is, data which could tell senior managers at T.Co how long clients were taking to get through each stage and into a new job. The project was guided by the Managing Director's assertion "If you can't measure it, you can't improve it" (Report: Useful Statistics for Outplacement, T.Co).

Furthermore, a senior consultant concurred, stating that sponsor organisations needed realistic measures of job search effectiveness, clients needed realistic and measurable targets and consultants needed to know how they were doing at each stage. He also said that he could use the system to work out what he was earning on an hourly basis.

5.5.2 The Sales and Marketing Wish List

In April 2000, the Sales Manager and two supervisors were involved in a further requirements gathering exercise. The purpose of this was to create a 'wish list' of

support mechanisms that would assist in reducing workloads. The wish list was as follows:

- The implementation of an enterprise-wide electronic diary
- Tweaks to the Corporate Filemaker Pro Database to allow improved views of existing data, the addition of triggers to generate sales appointment letters automatically from data entered, the ability to track the success of a mail shot and one button marketing pack creation and printing.
- Tweaks to the New Client Filemaker Pro Database including the provision of extra reference information so that enquiries could be dealt with more efficiently and the addition of a trigger to prompt contact of successfully placed clients 3 months after placement

Some of the requests were implemented and some were left until the 'new' system was put in place.

Table 5.4: Client Journey: Model I

Week	Client Consultant Meeting	Progress
0	1st Meeting (extended session)	Data gathered to begin drafting CV Finalise CV ready to be proffered Discuss responding to adverts Discuss speculative Approaches Research Brief completed (1 day turnaround) Complete draft speculative and recruitment consultant letters Compete Jobline 500 brief if required or demonstrate use to client Put structure into campaign – home office, methods of communication, job search newspapers, and library research.
1	2nd Meeting	CV approved Headline research approved Letters approved Review client networking list Utilise Jobline 500 Respond to advertised positions Conduct internet job site familiarisation Place on executive register at Job Centre as appropriate
2	3rd Meeting	1st TMP mail shot CV sent to recruitment consultants CV sent to network of former T.Co clients from same business sector CV posted on T.Co website for access by HR personnel CV sent to network of HR personnel
	4th Meeting	Go through 100 POC model Review results of 1st TMP Complete research brief to 2nd TMP as necessary Book interview training as appropriate
5		2nd TMP mail shot
7	5th Meeting	Go through 100 POC model Review results of 2nd TMP Complete research brief for 3rd TMP as necessary Book interview training as appropriate
8		Conduct Interview training
13	Major Review	Collation of client campaign statistics Review interviews attended to date Review replies and respond as appropriate
16		Average success time in 1999. Laggard clients – establish 1-2 month job placement at no extra cost to sponsor.

5.5.3 The 'Client-tracking' Project

By November 2000, when the TCS programme began, the Managing Director had decided that the company needed to migrate to the Windows environment and therefore all projects other than this were mostly put on hold. However, in November, the IT Manager and the academic team began documenting the processes at T.Co in preparation for a more detailed requirements gathering process for the client-tracking project. This led to a basic process map that was used to identify how

existing applications were being used. This process map was also used to guide data requirements gathering processes whereby users of the various applications and operators of the processes were interviewed about the status quo. Also:

- Many users had detailed their requirements by documenting the processes and procedures they worked with.
- Requirements were solicited from the operations manager.
- Various other process maps were also produced which had much greater detail[26].

5.5.4 The Consultants' Wish List

In January 2001, following the internal requirements gathering process, a meeting was arranged to elicit the requirements of the Consultants who worked for T.Co. This involved the IT Manager, the Operations Manager, a non-executive director, the Managing Director, the Sales Manager, the Research Manager, the Finance officer, the Managing Consultant (the spokesperson for the consultants) and myself. At the meeting, the managing consultant presented a further 'wish list' and a number of questions. The initial questions were:

- Why do we need a computerised tracking system?
- For whose benefit is it?
- What is the primary aim of the system?
- If it is primarily for the marketing people, how much extra work is required of the consultants to make it useful?

[26] Requirements documentation (Documents 10-14).

They concluded their list of questions with the statement:

> "There is some potential benefit for consultants and clients if we can collect quantitative data on the outcome of mailing shots, the routes to new jobs, and subsequent follow up of clients when they leave us; however, the benefit must make the extra work worth while." (Digest of "Wish List" of T.Co Northern 1:1 Consultants - Doc 19)

The non-executive director was quick to respond admitting that originally the project had been viewed as a cost reduction exercise. However, he went on to say that:

> "….following consultation, we have realised that we could not expect you to buy into this. Really the project is about improving client satisfaction. We don't want you to view the project as a system for monitoring and controlling you. We know you wouldn't buy into it on these grounds – it has to work for you, as well as us".

He then went on to argue that T.Co did need the system due to their rapid expansion. He said that without the system they would not be efficient and would therefore not be able to compete. He further built up his argument by reporting that on 12 January, they had 100 clients on the books and they expected this number to increase to 150 when the Midlands office took off. The operations manager also contributed at this point. He said that "Consultants get pestered by clients for details about the progress of their service, e.g. mail shots. Consultants feel it's not their job". The non-executive took over again and said really the system was about consultants tracking T.Co and ensuring that they did what they were supposed to. The consultants were not approached again regarding their requirements.

5.5.5 The Client-tracking Project Begins Proper

A new Operations Manager was appointed in June 2001. His role, according to the Managing Director and the company newsletter, was to "implement leading edge

systems and communications across the T.Co branch network" and "provide the technology to underpin the company's rapid expansion".

The new Operations Manager had extensive experience of implementing enterprise wide information systems projects in blue-chip companies. This manager came to the project with good ideas about the way forward and quickly became a respected part of the team. Several members of staff viewed him as a breath of fresh air. He had been told, by the Managing Director, that the new client-tracking system had to be in place by February 2002 and that this was his responsibility. When the project was launched, it was widely agreed that the implementation should begin in the research department, which was considered the most complex business function. As a result, end-users were made aware that a new software installation was planned for the future and they viewed this as a panacea to their problems. Quickly after it was publicised the IT Manager told me in a telephone conversation that staff were saying things like 'the client-tracking system will take care of that' and 'when the client-tracking system comes, my head will stop spinning'. However, the IT Manager was aware that whichever product was eventually selected would entail a trade-off between an ideal system and the constraints of a standardised package and so in an attempt to manage user expectations, she was keen to point this out on many occasions.

During the initial meetings (comprising the IT Manager, Operations Manager, Graduate Trainee, Research Manager and two academics), the academics advised the other members of the Project Management Team of the importance of eliciting and articulating user requirements to enable a more informed evaluation of packaged

software products and help narrow down the options. Whilst team members were accepting of this suggestion, minimal effort was put into consulting end-users. However, in order to aid their own understanding of user requirements, the Graduate Trainee and IT Manager conducted a further analysis of the client journey. During this same period, a focus day with end-users was scheduled on a number of occasions. It was hoped that this would provide feedback on the project team's understanding of user requirements and offer suggestions on future direction. The focus day never happened as staff were deemed to be too busy by their managers: "we'd love to get people involved, but we just done have the time" (Supervisor), "our staff are just too busy" (Sales Manager) or themselves "it's the wrong time of the month for me" (Finance Officer).

In addition, much of the previous requirements gathering processes were brought into use with the idea of creating a 'definitive' requirements document that could be used to assess various packaged software products, by now the favoured option over developing the Filemaker Pro applications any further. Various members of the project team (especially the academics) had advised the Managing Director that custom development would be costly, although not impossible, in the light of the emerging requirements. The accuracy of this document was later verified by two end-users. The main concern of most of the project team seemed to lie with ensuring the (financial) support of senior management. This was confirmed with much of the documentation that was produced that appealed to the interests of senior management. These documents included statements declaring "Our aim is to introduce a flexible system that will streamline and improve our current business processes and speed up the client journey thus becoming more cost effective."

Similarly, the client-tracking project was claimed to enable "T.Co to continue to provide a business class service and grow effectively in the future, whilst maintaining efficiency in all areas."

By November 2001, the operations manager had left, but a statement of requirements had been prepared. The main thrust of the requirements at this stage were:

> "T.Co are looking for an 'Off the Shelf' product that has excellent after Sales Support because of the geographical spread of the users. In addition, the remote users should be able to access the Client-tracking System effectively using a 54k modem, so having Intranet and Internet capability. The application should be compatible with our current infrastructure and be able to relate to the in-house Filemaker Pro Database and Sage financial database.
>
> The Client-tracking is a Workflow application that will enable the monitoring of the journey of a client as they travel through T.Co. Different activities will be triggered at certain stages thus creating an effective and efficient journey reducing time-spans and streamlining processes.
>
> Data that is collected throughout both stages of the journey can then be used in an intelligent manner for business decisions and business procedures at all levels. The application should have excellent action reporting capabilities, with the ability to present business knowledge, performance statistics, Key Performance Indicators that is user friendly and offered via a digital dashboard." (T.Co Statement of Requirements, Document 38).

5.5.6 Product Identification and Selection

Once the requirements process had been 'finalised', research was conducted into a variety of packaged software products so that a number of potential vendors could be short-listed. By December 2001, four potential products, from four vendors, had been identified via a charitable organisation, which assisted small companies in this kind of activity. This was a difficult task as the IT Manager told me they had been inundated with calls from vendors following their expression of interest. Each

vendor was sent the requirements document and process maps along with the invitation to tender for the work. Immediately, the provider of Siebel (vendor D) responded stating that they could not meet the company's requirements. This was interesting as the package was a market leader and was very sophisticated. Further probing about the reasons why they could not meet the company's requirements revealed it was more to do with the product being 'too big' and too costly – Vendor D had basically said 'you couldn't afford us' to the IT Manager. However, initial negotiations were set up with the three other vendors and the project management team.

The vendors included: Vendor A who supplied a Sage product; Vendor B who supplied Goldmine; and, Vendor C who supplied a product called Commence. Communications with Vendor A were problematic from the outset (reportedly a poor response on their part) and the product seemed comparatively expensive to other packages. Consequently, this company never went beyond initial negotiations with the project management team. Vendor C presented at T.Co and whilst the product was seen as flexible and within budget, it was not perceived as containing the most basic functionality that the company required and this avenue was therefore not progressed any further. Vendor B, who sold the Goldmine product, had a number of meetings with the project management team before demonstrating the product and discussing its capabilities with the Managing Director. Despite having carried out fairly detailed discussions on the nature of the company requirements and having viewed the supporting documentation, this presentation was unsuccessful in that the salesperson simply demonstrated the standard product and paid no attention to the needs of T.Co. The Managing Director concluded: "Goldmine isn't for us."

Following this, the Board expressed their discontent and noted their concerns about the value of a CRM package. They demanded more research into the possibilities of further developing their existing Filemaker Pro applications. The project management team were therefore forced to investigate the potential for custom development even though they did not agree with this strategy. In January 2002, the IT Manager reported:

> "Filemaker Pro is currently being evaluated as a viable option, as little work has previously been conducted to fully explore its full capabilities." (Client-tracking Update Memo - Document 41)

and

> "Filemaker Pro developers have been contacted directly to gain an appreciation of what the software is capable of. Recently, it has been established that there is a link to Sage, Outlook can be integrated and alerts and prompts can become automated. The next process is to create a list of critical specification that T.Co require from the Client-tracking System. This list will then be sent to our Filemaker Pro contacts so they can offer detailed feedback. A number of site visits will also be arranged so that SF and MG can gain a further insight into how developers can help T.Co." (Client-tracking Update Memo - Document 41)

However, the project team believed that a package was the best way forward and therefore had continued their search for a suitable vendor. The IT Manager was then approached by another vendor of the Goldmine product – Vendor E. A presentation was arranged, but this did not include the Managing Director. The project team were impressed by the level of personalisation that Vendor E had incorporated into the demonstration and further guidance about T.Co requirements were given by the team to the vendor throughout this. A further Vendor E product presentation was then arranged, but with the Managing Director involved. At the next presentation, this vendor made extensive use of the background information that they had accumulated and as a result further personalised much of the product terminology for their

presentation. This was well received by the Board. The Managing Director took control in this meeting and wrote a list of business functions on the board.

- Can it tell us where a client is in the process?
- Can we obtain management information about client progress that we can use to deal with sponsors?
- Can we obtain performance metrics to manage the process?

He then asked the vendors if the Goldmine application could support these queries – he ticked off each one as they responded. The application was reported as being able to do everything T.Co wanted. The Managing Director was delighted with the presentation and the responses to his questions. He immediately shifted position from his initial suspicion of the product to completely embracing it. He said "This system can do all we need, and more!"

All ideas of entering into custom development were brushed aside as quickly as they had appeared. The Managing Director also now changed his mind and stated that rather than install the CRM package in the research department with the possibility of future implementation across the other business functions; the system was to be installed incrementally throughout the whole organisation. However, although the actions of the Managing Director were interpreted in a similar fashion by the IT manager (in my discussions with her), it was reported somewhat differently at a meeting and in an executive summary she had prepared for this.

She wrote:

> "After a disastrous first demonstration from a selected vendor the project took a sideward step. There was hesitation in the decision of investing funds and resources into a new application or developing the current information system. The question to 'build or buy' an application really needed to be resolved so an investigation exercise was undertook to really understand the capabilities of Filemaker Pro. The investigation involved discussions and meeting with software developers throughout the UK. To finally 'close' the option the conclusion was that Filemaker Pro would not support the growth and geographical spread that T.Co aim for long-term. The option to buy and simply customise a product that will expand with T.Co was the favourable option. An excellent demonstration of Goldmine introduced a totally new holistic assessment of T.Co information systems and the observation that business functions do not exist in isolation became apparent." (Client-tracking Meeting Executive Summary - doc 46)

Following the successful sales pitch, senior management resistance to cost also seemed no longer relevant as the number of user licences increased and the costs were revised to £50,000, over double the original pared down estimates (following the Migration project budget realignment the project team had been told they had to bring the client-tracking project in at £25,000). However, requirements analysis had not been undertaken in any detail in respect of the processes requiring support in areas such as finance and sales and marketing, nor had those users been consulted.

5.5.7 Phase 1 of Implementation Planning

Interestingly, with the change in the roll out of the package from the research department to the rest of the enterprise – the starting point was altered too. The vendors felt that as the research department (and Client-tracking functionality) was the most complicated business function, this would be best left until last and they proposed a different phasing of the implementation process (Vendor E workflow document). This is shown in Figure 5.4.

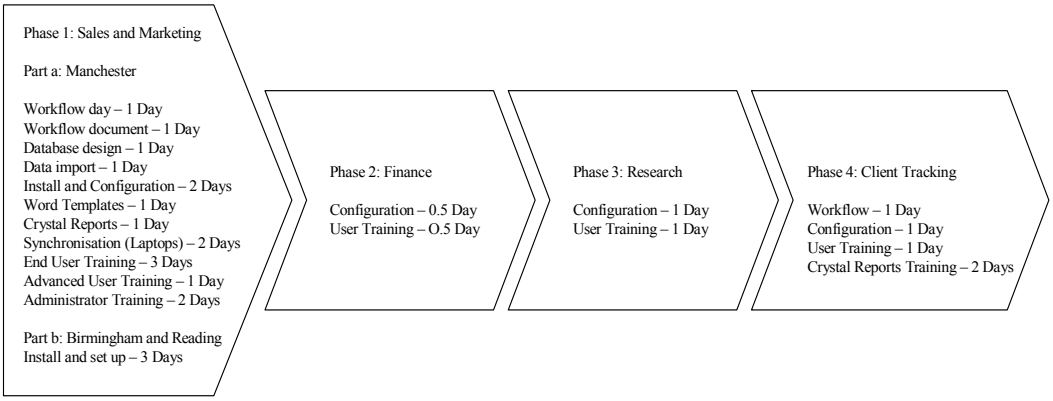


Figure 5.4:The Client-tracking Implementation Plan

It can be seen that Sales and Marketing were pushed to the front of the implementation as this was proposed as having the 'best fit' with the packaged software on offer. Interestingly, out of the costs for each phase, this was the most expensive, taking up nearly 60 per cent of the budget, the remaining phases accounting for 8, 12 and 20 percent respectively. Furthermore, the managing director quickly took credit and deemed it "an implementation strategy that allowed for risk management." The IT manager affirmed this in an executive summary later:

> "An intelligent decision was made in how the project was approached in an incremental manner dealing with individual functions and gathering expertise and skills along the journey and then applying that knowledge at the next stage." (Client-tracking Meeting Executive Summary - doc 46)

The process of implementation was set to begin 27 May 2002. From this point and up to the 26 June (when phase 1 as described above would begin), the company took time to plan. Key to this was the involvement of users, the IT Manager reported "Organisational change will be managed as a high priority and emphasis will be placed upon bringing the users fully into the project" (Client-tracking Meeting Executive Summary - doc 46).

It was communicated to certain users that they would be involved in the workflow days and that this needed to be prepared for. It was decided by the Managing Director that 'key people' from each of the business functions would assist by offering requirements and outlining procedures. The aim was to produce a portfolio for each of the four phases to include current processes, wish lists, templates and reports. The IT Manager told those involved that they had already received substantial advice from the vendor consultant and account managers. They were described as very approachable and she said they had offered excellent guidance so far. The planning process was ongoing when the first workflow day took place, more users got involved, and the project stalled.

5.5.8 The Workflow Day

The first workflow day was held, as planned, on 26 June. Each member of the project steering committee (the 'key people') was charged with voicing the opinions and requirements of their respective teams. The steering committee had rapidly evolved from the initial project team in the light of the re-scoping of the project. The Sales Manager, Marketing Manager and Human Resources Manager and Finance Officer were now included. The steering committee agreed that all T.Co personnel needed to have the opportunity to be included in the project to ensure minimum resistance to change[27]. The day was organised to begin with an initial full committee meeting and then sub meetings for the various phases which would include the various members of the steering committee as necessary. The Consultant from Vendor E began the meeting with an introduction to his company and the package and outlining the purpose of the day as getting an overall specification for T.Co. He

[27] Note that I, and the other academic involved in the project did not agree with this rationale. If users were to be included, we felt it should be so that they brought something to the process.

was quick to point out within the first 20 minutes that although the package was highly customisable "sometimes the organisation has to bend toward the product as well" and " you may transfer work practices from other organisations". The meeting then moved on to looking at the product. The Consultant said that it was the groups decision as to how they wanted to work with the product and pressed the point that if "you don't say it, you don't get it".

The Consultant asked for the committee's requirements. This led to around twenty minutes of various members of the steering committee articulating their requirements in a rather un-orderly fashion. I intervened at this stage and suggested it might be a good idea to start at the beginning of the process. This meant starting with the generation of sales. The discussion then proceeded with various staff illustrating the journey. The Consultant would then configure the package in front of the staff (a projector had been linked to his laptop). As this progressed, T.Co staff refined and generated further requirements as they began to see what the application was capable of. Where requirements could not be solved by configuration, the Consultant made notes to investigate this further. It was very clear that the Consultant had not familiarised himself with the documentation sent to his company, as part of the tendering process as he had no concept of the basic workings of T.Co in terms of processes or terminology. For example, the nature of the business, the role of consultants, the nature of sponsors and of clients had to be explained to him by various members of the committee which frustrated them. At a coffee break, I asked the Research Manager, Sales Manager and Human Resources Manager how they thought things were going. As the quotes below demonstrate, they were worried about he capabilities of the Consultant and the package.

> "He hasn't read the documentation [T.Co's requirements specification], he could have least have done that." (Research Manager)
>
> "I hope this is going to get better, it's a bit of a waste of time at the moment. Is Goldmine going to do what we want?" (Sales Manager)
>
> "I've only just joined the company and I know more than he does, he's just not prepared." (Human Resources Manager)

Table 5.5 details some of the 'highlights' from the meeting.

Table 5.5: Highlights from Workflow Day 1.

The Sales Manager wanted to be able to convert a client into a sponsor – it was discovered that Goldmine could not do this. A new record had to be created and this meant that the history regarding the sponsor (as a client) would be lost. She did ask if Goldmine would ever be able to do this and the consultant said, yes, if enough customers asked for it.
The Research Manager virtually swooned when she was offered the pipeline functionality in Goldmine for client-tracking purposes. However, it was not possible to construct individual pipelines that reflected an individual clients progress. Nor was it possible to create various standard pipelines which reflected various T.Co products and the points at which clients would normally be expected to have done things by as shown earlier in
Every member of the team got involved in some form of selling of the package in terms of trying to make it more appealing to T.Co. In effect, we helped the Consultant sell his product to us. For example, there were many occasions where staff agreed to change ways that they worked because the system could not facilitate the processes as they stood at T.Co at present.
The Sales Manager wanted reminders to be set automatically for follow up actions. That is for example, if a brochure was sent to a potential sponsor – a reminder would be set to follow up with a telephone call 7 days later. The Consultants initial position was that this was not possible, but I knew from a previous implementation that I had been involved with that it was. I asked the Consultant about this and he said it would take more time to configure the software – but agreed to do it.
The Consultant interpreted the staff being very specific about their requirements as 'naval gazing', stating that they were getting 'into the detail'. He also took a 'U' turn in that he said the purpose of the day was to focus upon Sales, not other areas of T.Co.

By the end of the day, staff still felt very uneasy about the selection of Goldmine and these concerns were reported to the Managing Director. An email exchange also took place a few days after. The Managing Director had contacted Vendor E to express his concerns and had been advised to wait for the workflow document to be delivered. The Managing Director had also asked the IT Manager to write to the

Consultant at Vendor E outlining the issues. Before the letter was sent it was passed by the Managing Director who watered it down and sent it as an email himself. The email stated that he was disappointed as he thought that he had determined that Goldmine was the right product. Furthermore, he had assumed the workflow day would be about aligning T.Co processes with those embedded within Goldmine, rather than working out if Goldmine was the right product; and that he would wait for the documentation to be delivered before deciding whether to proceed. He then sent a further email to the members of the steering committee. In this email, the Managing Director made several points:

- He felt the problems had been caused by the attitude of the consultant.
- He was convinced that Sage and Goldmine would work together.
- Data cleansing was a necessity
- The consultant was the only project manager Vendor E had so they would have to work with him
- He suspected that the salesperson that came before had raised 'our' expectations; possibly, to unreasonable levels and that the Consultant had possibly moved too far the other way.

I responded in broad agreement with the Managing Director although I added four points:

- Data cleansing was not just about inaccurate data entry; it was mostly because they had eight pools of overlapping data.

- The Consultant did not get up to speed with how T.Co does business so he could not show the benefits of Goldmine in T.Co terms.
- Because of this, his focus upon getting T.Co to bend to match the software aggravated the situation as staff felt they had to change to something that was inferior. I pointed out that the staff had been very positive about learning new ways and working with the package to start with.
- In order to find out about the package, the team had found it necessary to report their requirements but the Consultant had interpreted this as getting into too much detail. I stated that this was very important to ensure that T.Co got the most from the finite functionality available. I also pointed out that, when pushed, the Consultant revealed functionality that he said was not available and suggested that Vendor E may want to implement the most basic model of the package with minimal configuration.

The IT Manager also responded to the email reinforcing the need for the data clean up and the ability of T.Co to develop a good working relationship with the Consultant and Vendor E. This exchange was followed by a meeting of the project steering committee, with the Managing Director, on the 5th of July. The purpose was to determine the best way forward. The main thrust of this meeting was the Managing Director got people to confirm that Goldmine could 'broadly' do what T.Co required. For example, at one point the Managing Director said "…we know there are problems with Goldmine, but can it do most of what we want – yes or no?"

The Managing Director really wanted a yes or no answer. The majority of the committee therefore acquiesced. On this basis, the decision to proceed with

Goldmine was made. However, this was not communicated to Vendor E, the Managing Director wanted to be completely sure Goldmine was the right product and wished to wait for the workflow document to arrive.

5.5.9 Phase 2 of Implementation Planning

It had become clear in Phase 1 of the implementation planning process that the company needed to plan for a data cleansing exercise and the implementation of data entry standards. The need for clean data and the maintenance of clean data throughout T.Co was argued to be due to the implementation of Goldmine. One document[28] states "in preparation for the Goldmine project, our existing data will have to be cleansed and constraints put in place to prevent data becoming 'dirty' again".

It was argued that the new system would create a single store of data with all departments sharing this, meaning that everyone needed to keep to the same "rules". The document itself was aimed at identifying and eliminating areas of ambiguity in T.Co data entry. However, the team did offer the document up for consultation. A selection of the proposed data entry standards is shown below (underlined words show the research departments preferences):

- LTD, Ltd Limited
- PLC, plc. Plc (this is still an area of uncertainty)
- Road or Rd, Street or St, Avenue or Ave.
- Mgr, Manager

[28] Goldmine Data Quality - document 48. Authorship: IT Manager, Research Manager, Graduate Trainee and Sale Manager.

A particularly interesting one was the use of salutations "Dear Sirs or Dear Sir – there are many women in senior roles, especially HR, will this greeting insult/annoy them?" The consultation document ended with the following:

> "The database is going to be used by YOU, therefore your comments and opinions are extremely important. Please reply via email to (C) & (M) by Wednesday 17 July and we will draw up a formal data entry procedures document by Friday 26 July. This document will then be the official record of T.Co's data entry standards and will be strictly enforced by the 'data stewards'. Clean data are exceptionally important and we appreciate your co-operation." (doc 48)

However, data cleansing had not been factored into the timings of the plan by the vendors, yet it had to be carefully planned, as it had to happen quickly. The data cleansing activities would result in a total system shut down for Sales and Marketing. Therefore, I agreed with the IT Manager that data cleansing would happen from a Friday to a Monday with staff working over the weekend. Moreover, user testing had not been included in the plan as the assumption had been made that the data would transfer neatly in a day. Therefore, plans for user testing also had to be incorporated into the plan. In addition, I suggested that testing could not happen until training had been provided and therefore further changes had to be made to the schedule. It was however decided that training would not happen before testing. I also suggested that the 'go live' date might be a useful one to highlight. A comparison of the vendor and T.Co implementation plans are shown in Table 5.6.

Table 5.6: A Comparison of Vendor E and T.Co's Implementation Plans

Vendor Plan		T.Co Plan	
Sales and Marketing (a)		Sales and Marketing (a)	
Sept 30 Oct 21 23 24 28 29 31 Nov 4 7	 Workflow Document to T.Co T.Co Agrees Workflow Configuration of Goldmine Installation of Goldmine Data Import End User Training Advanced Training Admin 1 Training Admin 2 Training	Sept 30 Oct 21 23 24 25 28 29 30 31 Nov 4 7 13 14	 Workflow Document to T.Co T.Co Agrees Workflow Configuration of Goldmine Installation of Goldmine Data Cleansing Data Import End User Training 'Go Live' Advanced Training Admin 1 Training Admin 2 Training Crystal Reports Training Crystal Reports Training
Sales and Marketing (b)		Sales and Marketing (b)	
Nov 25 26 Dec 2 3 4	 Installation of Goldmine (Laptops) Installation of Goldmine (Reading) End User Training End User Training End User Training	Nov 25 26 Dec 2 3 4	 Installation of Goldmine (Laptops) Installation of Goldmine (Reading) End User Training End User Training End User Training
Research		Research and Client-tracking had yet to be reviewed.	
Dec 5 9 10	 Configuration of Goldmine End User Training Advanced Training		
Client-tracking			
Dec 11 12 16	 Development of Crystal Reports Development of Crystal Reports End User Training		
Finance			
After Christmas			

5.5.10 Signing Off the Workflow Document

The first workflow document arrived mid July and it did not meet with the expectations of the members of the steering committee despite the workflow day. The Research Manager said "it does not provide us with enough detail about the

proposed system for us to sign this off". The IT Manager was similarly unconvinced stating that "it's not clear what we are buying at this stage, it's going to need more work", and when I asked the Sales Manager what she thought, she said "I have been busy, and there's a lot to get through, it looks okay but how can we be sure?"

Further meetings were then held within T.Co (but not involving the vendor), to 'double confirm' that Goldmine was the right product. Another meeting was then held mid August with the Sales Person, Account Manager, Managing Director and Consultant of Vendor E, to determine if T.Co could work with them and how Goldmine could be made to work for the company. The IT Manager informed me that it was agreed at this meeting that T.Co and Vendor E could work together. She said the Managing Director had pulled off his favourite 'one-time party trick', which he did with all vendors, whereby he threatened to withdraw from the deal if he didn't get what he wanted. In this case, a commitment from Vendor E to deliver the system they had promised in the sales pitch.

Regarding functionality, the IT Manager and the Graduate Trainee had been charged with the responsibility of working through T.Co requirements with the Consultant the morning before the afternoon meeting. However, when the meeting was held, the Managing Director stated he had different, simpler requirements and outlined these. Essentially, he did not want the system to have 'triggers' which prompted or launched activities. Instead, he wanted the package to be able to generate exception reports that would highlight those areas where deadlines had not been met. For example, staff would not be prompted to chase a missing CV 'by' the package – instead an exception report would be printed out and given to the consultant to

highlight which clients they were working with that had CVs outstanding. In the first workflow day and the workflow document, the original 'trigger' based requirements had been 'bent' around a piece of sales functionality for pipeline management. The Managing Director's interventions freed up this piece of functionality for sales. This change was reflected in a second workflow document that arrived at T.Co at the end of September. The agreement of the workflow document was re-scheduled for 21 October 2002 but a further internal meeting was arranged for 27 September 2002 in order to ensure that T.Co requirements were incorporated into a further revised workflow document, as this was still not meeting the requirements of the steering committee.

By 8 October the purchase of the package had been postponed to the beginning of December and by November, this had moved to February 2003. Late in November, I interviewed the IT Manager about the progress of the requirements gathering process and the workflow document. She said that it had not been agreed and it was becoming difficult to keep staff involved because of the postponements and false starts. However, the Research Manager was keen to get the package implemented and she still felt that the new workflow document needed further attention. Specifically, she wanted to see screenshots of the package and more detail about functionality. The IT Manager said "This isn't over, I expect the workflow document to be double the size it is now – just you see."

5.6 Summary Review

This case describes the contexts of three projects that offer insights into a variety of issues associated with packaged software but especially those associated with

selection activities. By necessity, I have had to tell the three stories within themselves in order to keep things clear. However, in summary it is useful to bring together the highlights from the three stories and illustrate the overlaps amongst them (Figure 5.5). The next chapter presents the analysis and findings from the data.

Figure 5.5: Summary Timeline of Events at T.Co.

6 An Analysis of Packaged Software Selection Processes at T.Co

6.1 Introduction

This chapter presents the analysis of the findings of the field study. The aim of the analysis is twofold. First, it demonstrates the problems associated with rationalistic theories of the packaged software selection process. Second, by doing this, it is possible to gain insights that offer a useful basis for developing an alternative theory of packaged software selection. In order to do this, the conceptual framework presented at the end of chapter 3 is used to structure the analysis - see the copy in Figure 6.1.

The chapter begins with a discussion of the reasons why the various packages were chosen by those at T.Co. The various processes of selection that were enacted are then considered, using the ideal model of selection as a guide. This involves examining the user requirements gathering processes, the processes of package evaluation and the final selection decisions. The implementation and post implementation issues arising from the cases in relation to the respective selection

processes are then discussed. Those issues associated with the market environment are raised as necessary throughout.

It is important to note at this point that the analysis presented here is not meant to be an exhaustive interpretation of the complexities of packaged software selection; other studies would no doubt reveal different things. However, the data are sufficient to demonstrate the problems of rationalistic theories of selection reported in the literature and provide a basis for the development of an alternative theory.

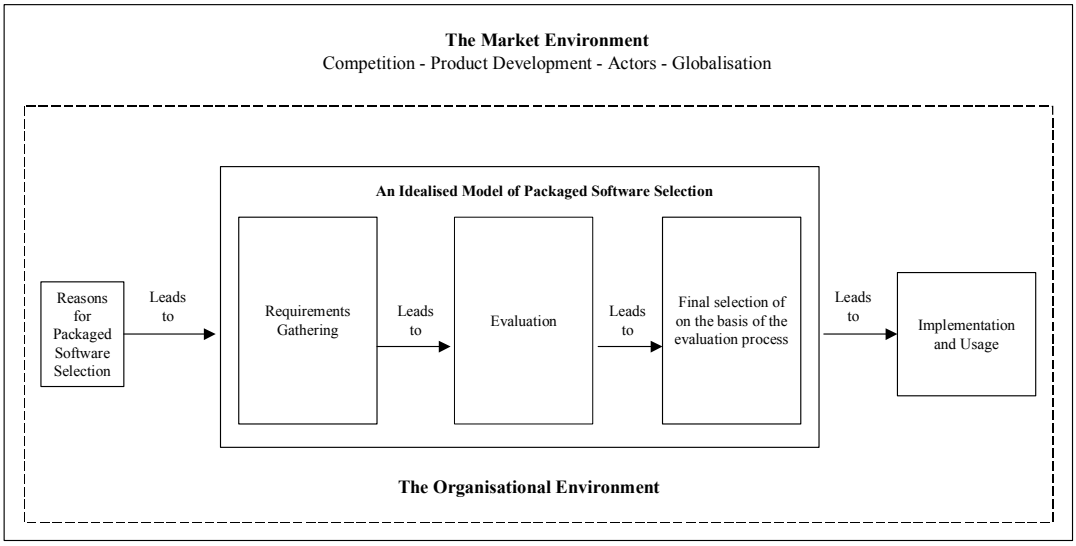


Figure 6.1: The Conceptual Framework

6.2 Reasons for Packaged Software Selection

In this section, I draw from the cases the reasons why packages were chosen over custom development. These reasons relate to the:

- desire to standardise and attain 'best practices';
- need to 'deal with' legacy information systems problems
- need to 'deal with' an applications backlog;

- role of selling;
- cost;
- wish to 'buy into' a tried and tested solution with accompanying readily available knowledge and skills base;
- possibility of outsourcing maintenance and development;
- implementation of organisational change;
- role of bravado.

The forthcoming analysis demonstrates the complexity of decision making in relation to packaged software adoption, particularly with regard to how reasons can be variously interpreted as 'irrational' and 'rational' because of the diversity of perspectives involved. It also infers that reasons for adoption might emerge and change before, during and after the selection process is enacted.

6.2.1 Standardisation and Best Practices

T.Co clearly had ambitions of standardisation in relation to the various packages they implemented. This standardisation fell into two categories, buying into a standard that allowed them to more readily work with others and buying into standard ways of working.

The first form of standardisation, to facilitate working with others, is emphasised early in the migration project. The problems encountered with running the Mac version of Office and Email are clear. Staff could not readily work using electronic means with sponsors, clients and each other. For example, access to email was limited to only a few members of staff. More importantly, electronic documents

often became corrupted as they were transferred between the Apple Mac and Microsoft environments even though this was usually between the same package. For example, a CV created in Word for Apple Mac being sent to, or received from, someone who used Word for Windows. Personnel at T.Co made it very clear that they needed to adopt the industry standard packages but, critically, they had to use them on the industry standard platform (Windows). The Managing Director had previously been unsure of moving to the Windows platform, but his reasoning changed when an external IT Director, who he saw as an 'expert', told him it was a "no brainer". Essentially, the perception of the IT Director was that there was no alternative and this was then shared by the Managing Director.

Various T.Co personnel also wanted to adhere to what they saw as standard ways of working. Again, in the migration project, the move to the Windows version of the Sage application allowed them to have one invoicing system where previously they had used three. The Managing Director was quoted as saying " it's ridiculous, we must be the only company in the UK with three invoicing systems, Sage gives us the opportunity for one – the industry standard". Broadly speaking, the Managing Director saw Sage as the most efficient choice to make.

More dramatically however, the exploitation of standard ways of working became a significant aspect of the client-tracking project. At the workflow day for example, the Human Resources Manager made the point that T.Co did not want to 'reinvent the wheel' and suggested that there must be generic codes for sales and marketing purposes the other people had used. She really did want to see other organisations' configured packages and use them where appropriate. Again, as with the Sage

application, it can be seen that there was a desire to make a decision base upon the criterion of efficiency.

Personnel at T.Co aimed to implement packaged software in order to improve communication throughout the organisation in a similar fashion to those cases reported in the literature (Holland and Light, 1999a; Ross, 1999). The T.Co case also adds further detail to this idea in that a reason for the purchase of packaged software was to improve communication capabilities with those external to the formal organisational boundary, particularly clients and sponsors. Indeed this was a major reason for the migration project involving the move to the standards of the Windows environment and mainstream applications such as Microsoft Office. Therefore, not only were the decisions made on the grounds of efficiency, the decisions were supposed to have positive implications for the effectiveness of T.Co.

Moreover, the Managing Director and one of the Non-Executive Directors took this further, believing that the client-tracking system would allow them to exploit state of the art technologies and benefit from industry-wide standards of efficiency if they bent a little towards the package. The last comment indicates a desire for more than standards for communications purposes – T.Co were interested in standards that were perceived as 'best in class'. Yet, they felt that it was their processes that were the best in the sector, as they were more personalised. Therefore, although they might have been perceived as more efficient, T.Co has decided not to use some of the functionality of the Goldmine package as it would have, as the Managing Director put it, "watered down the client experience". Further probing revealed that really the 'best in class' to which they aspired, was in relation to common data

standards, and non-value adding administrative processes where they felt they were inefficient due to the way they had evolved their Filemaker Pro applications - the TMP associated processes, for example.

The case of client-tracking adds insights into the argument that best practices may be inscribed into packaged software (PriceWaterhouse, 1996; KPMG, 1998). For personnel at T.Co the so-called best practices embedded in the package they wanted to attain related to non-value adding processes. Indeed, the package had the potential to erode their differentiation strategy of personal service, but they were very keen to ensure that any automation that did occur had no impact upon this even though it may have been viewed by other organisations as best practice. This adds to the way in which the idea of best practices is treated in the literature as it highlights that this is subject to interpretation. Therefore, because the benefits of best practices may be context specific, any so-called best practices embedded in packaged software would need to be evaluated rather than unconditionally accepted as universally advantageous as was done in for the music industry at 'Global Entertainment' (Light et al., 2000).

In summary, T.Co's experiences in respect of standardisation suggest that the utility of 'best practices' is subject to interpretation. Even where the standard nature of packaged software is purported to have benefits (such as increased efficiency), this may not necessarily be desired by organisations who may perceive their practices more favourably. Moreover, it becomes clearer that prior to the enactment of the selection process, vendor sales pitches which espouse the benefits of 'best practices' and standardisation, and such recommendations as those via colleagues or business

communities, may seem to offer a rational basis for proceeding with packaged software selection. However, it might not be until the process is enacted that the those in the consumer organisation realise that it would be inappropriate (and potentially hitherto perceived as irrational) for them to totally adhere to the standards and 'best practices' embodied in the software.

6.2.2 The T.Co Legacy Information Systems

T.Co had typical legacy information systems problems. The applications in place had evolved incrementally and in an unmanaged fashion. The applications had also been created by staff who had received no formal training in systems analysis, design and development. The result was a set of flat file databases that supported fragmented applications which duplicated data. Additionally, due to the nature of the applications and a lack of discipline in relation to data entry, much of the data the applications contained, and worked from, was inaccurate and formatted in a variety of different ways. Internally, the applications inhibited working across departments. Externally, this situation was amplified due to interoperability problems arising from the Apple Mac infrastructure. One aspect of the rationale for all of the packaged software purchase decisions relates back to these problems.

The Windows environment (a packaged operating system), the Microsoft Office and the Outlook applications were implemented to help overcome interoperability problems internally and externally. Despite these problems, there were, and still is in some areas, a great deal of attachment to the Apple Mac environment and the associated applications. For instance, when problems arose throughout the migration project and in the subsequent bedding in period, staff were quick to blame the

Windows environment and hark back to the halcyon days of the Apple Macs. Interestingly, many staff held the (obviously problematic), existing situation to be something to which the new windows environment and packages had to aspire. The IT Manager summed it up very well when she said that people were viewing the Apple Macs with 'rose tinted spectacles'.

T.Co's experiences suggest that the reasons for packaged software selection may not be universally agreed upon. For example, some would have preferred to stay with the Apple Mac environment. Additionally, it is shown that some arguments may be overruled essentially because they are seen as insignificant or irrational by others (in this case – senior management).

6.2.3 The Applications Backlog at T.Co

The capabilities to develop applications to support the expanding business and demanding functionality requirements were absent at T.Co. This can be seen as a form of applications backlog although those at T.Co might not have termed it as such. Essentially, the backlog comprised the various requirements although it was not located in any one place as, at the beginning of the project, there was no formal information systems function. Despite aspects of the literature relating to packaged software offering reasons for its uptake as resolving the applications backlog through reductions in development speed (Heikkila et al., 1991; PriceWaterhouse, 1996; Robson, 1997; Butler, 1999; Li, 1999), the T.Co case offers food for thought.

Certainly, this argument is borne out with the implementation of the Windows operating system, Microsoft Office, Outlook and OneSource applications.

Obviously, it would have been foolish for T.Co to attempt to build these applications themselves and it would definitely have taken them longer to do this than purchase them. Indeed, for OneSource it would have been nearly impossible as they were buying data as well as functionality. However, the client-tracking project reveals a situation that is less than clear cut. The idea of using a package for the client-tracking project was seriously raised and pursued from June 2001. However, it is important to note that a great deal of requirements gathering work had also been undertaken from as early as August 1999, although the company were not clear at this time about the buy – build decision. At the time data collection stopped (Early December 2002), the 'sign off' of the vendor's workflow document detailing the agreed configured package, selected in January 2002, had not happened and it does not look as though this will now happen until early in 2003. The applications backlog remains.

Clearly, some of the time taken to implement the client-tracking project may be attributable to the capabilities of the project team and other factors other than the package. However, this is the problem of certain previous views of packaged software selection. These views tend not to account for differences amongst implementation contexts and instead rely solely on the idea that because packaged software is built for mass markets, the economies of scale are bound to be achieved (Welke, 1981; Chau, 1994; Chau, 1995; KPMG, 1998). Yet, unlike products such as motor cars or kettles – where it would be undoubtedly more expensive to build to order, software may require much more effort to implement once it has been built. Indeed, it is highly plausible to assert that, in many cases, the only economies of scale that would readily be achievable are those of the packaged software vendor.

Thus, it does not matter if T.Co are lacking capabilities or if the requirements process has taken so long because they have high levels of information systems maturity which compel the project team to undertake a thorough job. The significant point is that, for whatever reason, the selection process is taking considerable time, even with requirements gathering work spanning well over two years. It is impossible to ascertain whether T.Co could have commissioned a custom developed application in this time and at what cost. Yet, the data presented here certainly raise questions about universally favourable implementation timescales for packaged software over custom development so frequently reported in the literature (cf. Heikkila et al., 1991; PriceWaterhouse, 1996; Li, 1999).

6.2.4 Forms of Selling

Two streams of selling activities are present in the T.Co case: the sale of the idea of a move from custom development to packages and that of choosing between various products and services. The analysis demonstrates the influence of the market and the agendas of non-market participants upon the reasons for packaged software adoption.

There was no sales activity in relation to migration, with regard to the move from custom to packaged software, due to the nature of the project. However, selling activity was clear in the external consultant's initial advice to the Managing Director about a suitable vendor, and then the various sales pitches that followed from others. In respect of OneSource, the 'sale' of the (partial) move from custom development to a particular product was enacted solely via the Research Manager. In both of these projects – various acts of selling are present. However, the role of acts of selling are

most strongly represented in the client-tracking project and thus, this shall now be focussed upon in greater detail.

The more usual selling activities associated with packaged software were clearly evident. Various channels were used by T.Co to identify potential 'client-tracking' products – this included the Internet and the trade press. Significantly, the use of a charitable 'dating agency' by T.Co led to an onslaught of approaches by organisations attempting to sell their product. These acts of selling clearly demonstrate how the process is far from one sided, and completely in the control of the consumer organisation, as the ideal model suggests. Various vendors attempted to convince those involved in the selection of the packages at T.Co that their product was the one that should be chosen. It was not simply a matter of personnel stipulating requirements and then choosing a product that most readily fitted with these. Additionally, the case of client-tracking further illustrates how the process of selection may not conform with the ideal model. Vendor D advised T.Co that their product would be too costly for them to implement. Although Vendor D's product would have met the requirements of T.Co, the vendor evaluated T.Co and halted any further discussion. The ideal model of selection does not take account of this.

Following the analysis of T.Co's existing applications portfolio and needs, T.Co were advised by the Academic Team that it was in the company's best interest to consider a packaged software strategy. This was accepted by the Managing Director initially. However, following problematic sales presentations for the client-tracking project, the Managing Director changed his mind believing that the potential of custom development had not been explored fully – a position which he maintains to

date. It really was the selling activity of Vendor B who were demonstrating the Goldmine product that motivated this shift. The vendor did not tailor the sales pitch to T.Co. This demonstrates how, even initially, although the reasons for packaged software had led to the selection process being enacted, the process raised further reasons for non-adoption and it was formally halted.

However, the project team were still interested in pursuing the packaged software route when they were approached by Vendor E, who sold the same product as Vendor B. This time, the project team vetted the first presentation without the Managing Director being present. The project team used this to assist Vendor E in further tailoring their sales presentation in line with the requirements of T.Co. This resulted in a highly personalised second sales pitch that captured the imagination of the Managing Director. Concurrently, the project team had managed to engineer the sale of the implementation of packaged software over a custom development and a previously unpalatable packaged software product. This was because they worked with the vendor to influence the decision making process of the Managing Director by ensuring that the sales pitch satisfied his problem. At the same time, it also satisfied the other problems of the staff (such as the Research Manager's workload) involved and Vendor E (who wanted to sell a product), which were not necessarily high on the agenda of the Managing Director. Thus, although there may well be different agendas behind the reasons for packaged software adoption, they may not always be in conflict. Moreover, this process demonstrates the role of the market in terms of the actor's capabilities to convince those in a consumer organisation to adopt packaged software. Vendor B and Vendor E offered the same product. Even

though Vendor E's offer ended up being marginally higher in cost than Vendor B's, they were selected as they were perceived to be capable of doing a more skilful job.

The highly personalised sales pitch did however, have repercussions. Following the difficulties of the workflow day, the Managing Director speculated that the salesperson may have raised expectations to unreasonable levels. Previous research suggests that sales presentations are only useful for showing 'bells and whistles' as the buying group may consist of representatives of many departments leaving the sales person with no option but to pitch it to the lowest common denominator (Gross and Ginzberg, 1984). The 'bells and whistles' approach won the contract, but then led to difficulties as the consultant attempted to curtail expectations in the workflow day where further reasons against adoption emerged as a result.

In summary, in addition to shedding light on the role of selling the idea of package adoption before selection is enacted, this discussion also demonstrates how this continues into the selection process. Moreover, these selling acts might result in further reasons for adoption, but critically reasons against adoption too. This further implies that just because the selection process begins, it does not mean that it will end with the implementation of a product.

6.2.5 Cost

Cost emerged as a consideration throughout the process, but not as might be expected. Lower cost is generally reported as a key decision factor in packaged software selection over custom development (Chau, 1995; PriceWaterhouse, 1996). However, this study suggests that as a reason for package adoption (and maybe even

custom development), the importance of cost may sometimes be overstated. The experiences of T.Co illustrate that although costs were a consideration it was not just a question of them being lower than custom development - they were sometimes accepted, rejected, increased and decreased.

The first instance of cost in the study relates to the migration project. Initially, the project was refused as it had instantly run over the budget of £50,000, a figure that was transplanted from the client-tracking estimate. More specifically, and once the initial overall costing 'problem' was dealt with by convincing the senior management team (in particular the Managing Director) of the case for investment, there were few queries about cost in relation to the Windows, Microsoft Office, Outlook and Sage packages. Interestingly, the Managing Director advocated increased costs in relation to the retention of the Filemaker Pro database package environment. It was suggested by the project team, that, as the Microsoft Office suite included the Access package, the Filemaker Pro applications could be moved to this (and improved in the process). This would have negated the need to purchase many Filemaker Pro Windows licences to replace the Apple Mac licences. It was however, decided by the Managing Director that to switch to Access was too risky and he preferred to increase expenditure instead. Yet again, efficiency became subservient in the decision making process and the Managing Director's view dominated.

Cost was also not regarded as a very important issue in relation to the OneSource project, despite this being relatively significant in the other two projects. This was probably the case as the costs were known up front. These did not change, as they

did in the other projects, because the implementation of the package was very simple. Moreover, the Research Manager had spent a good deal of time focussed upon priming the Managing Director and the rest of the Board for what was to come and convincing them of the multiple benefits of the purchase. Clearly there are parallels here with much of the information systems literature that espouses the importance of top management support (Pinto and Slevin, 1987; Lockett et al., 1991; Bashein et al., 1994). Moreover, the research function was seen as a critical function, which may explain the lack of resistance to the expenditure. There was also no real perceived alternative as the TMP application was proving inadequate and other packages were instantly dismissed by the Research Manager based on her previous experience.

With respect to the client-tracking project, cost was raised as an issue on multiple fronts. An initial issue in relation to cost was which packages T.Co could afford to purchase. A leading vendor in the marketplace instantly dismissed themselves from the T.Co search advising them that they could not afford such a package. Even if T.Co could have afforded the package, there might still have been questions on the vendor side in respect of whether it would have been cost effective to get involved in an implementation at a small company as their ability to realise their economies of scale would have been less.

As costs for the packages became clearer, the Managing Director felt that it would be cheaper to build a new system in the Filemaker Pro environment. This was because the existing systems had cost very little to develop and maintain in the past. Because of this, the Managing Director took a lot of convincing by the project team that

packaged software would be a better way forward. Indeed, in the end, the cost factor did not change his mind, it was the 'wow factor' of the sales demonstration of Vendor E that did so. Ironically, at this point, costs in relation to the project actually rose with the previously tight budget being doubled. This is not dissimilar to the concept of 'contagion' whereby as those in organisations begin to realise the 'benefits' of using IT, costs escalate dramatically in an uncontrolled fashion (Nolan, 1973; Gibson and Nolan, 1974).

The various packages discussed in the study demonstrate an important point. Although it has been suggested that the costs of package selection, implementation and usage can be predicted quite reliably (Golland, 1978; Heikkila et al., 1991), this study suggests the matter is more complex. Certainly, the licence fees associated with the packages purchased were a relatively standard matter, as were baseline maintenance agreements. However, this study also questions existing theories of packaged software which assert that they are universally easier to predict than custom development. If this is the case, then why did the migration project and client-tracking projects both go over the original estimates?

If the initial 'transplant' of the cost from the client-tracking project and to the migration project is put aside – there were still cost increases in relation to the package selection, implementation and usage processes. Part of the answer is that the number of licences increased due to company expansion. This also led to the need for a greater training budget. Additionally, more training had to be provided to existing staff as the packages brought with them new ways of usage which led to difficulties such as the Email RTF problem. With regard to the client-tracking

project, the original estimate of the project was shown to be sensible in the end. However, the Managing Director had previously slashed this budget with no good reason and then reinstated it after he saw the package in a different light with expanded coverage. Like the cases of Threads (Holland and Light, 1999a) and Fox Meyer (Bicknell, 1998), the overall costs for the packages were far more unpredictable. This is because the realities of organisational life (where decisions are made on a whim, inadequate information and where the situation may rapidly change) have to be taken into account.

6.2.6 The Tried and Tested Solution with an Associated Knowledge and Skills Base

Whilst the broad availability of a support network was important to T.Co, the robustness of the packages they considered was implicitly accepted. Moreover, even where evaluations of others' experiences with the packages were undertaken, this was mostly concerned with the nature of the support network.

T.Co had already experienced difficulties in supporting their custom applications, precisely because they were custom applications. Moreover, the development team came from in-house and were not information systems professionals. Therefore, T.Co had a very restricted avenue of support which ended with their Apple Mac support team who were technically oriented. If they had problems with the functionality of the software, there was nowhere to turn. Consequently, there was great appeal in purchasing a product that other people were using and that they could view before they adopted it. However, T.Co did not attempt to find out about the

quality of the production of the products themselves, as is suggested to be typically the case by (Sawyer, 2001).

With regard to the packages implemented as a result of the migration, the site visits were merely used to assess vendor capabilities. The functionality of the packages was in no way in question. The T.Co team 'knew' that Microsoft products were tried and tested solutions with a broad skills and knowledge base. They knew, for example, that they could recruit new staff and they would generally be familiar with the Windows environment and Microsoft Office applications (in contrast to the coverage of the Apple Mac environment where they had often had to train new staff).

In a similar vein, OneSource was viewed as a tried and tested solution because the Research Manager had used it in a previous role. Moreover, the case was made that this was widely used in other settings, adding to the idea of tapping into a broader knowledge base. However, the Research Manager's assessment of OneSource as 'the' product was never questioned.

In relation to the client-tracking project, like many organisations, T.Co undertook reference site visits which they found very useful for finding out about vendors and also, to some extent, about the packages on offer. However, the project team did not find the visits as useful in terms of assessing functionality capabilities as they claimed they operated very differently to the reference sites that they visited. In respect of the client-tracking project, the tried and tested solution idea for T.Co was very much about buying into a support network rather than viewing the package

selected as a solution that had worked elsewhere and would work for them. As the project progressed, it also became apparent to certain senior managers that, by selecting the package they chose, it might be possible in the future to advertise jobs in Sales for example, which asked for familiarity with Goldmine.

In summary, the experiences of T.Co suggest that the reason for buying a 'tried and tested' package might be concerned as much with buying into a support network and wider user base as it is with buying a well built product. Indeed at T.Co, the reliability of the construction of the packaged software products they implemented was taken for granted. Moreover, T.Co's experiences have demonstrated that this reason for adoption may emerge after the decision to adopt has been made.

6.2.7 Freeing up the Information Systems Related Staff

The T.Co study begins with no information systems staff presence in the organisation. Systems development at that time was user-led, with technical support provided externally. Therefore, there was no information systems staff group to release for other work. However, the Research Manager was engaged in any development work that was required and so she was the nearest equivalent. As the TCS programme got underway, however, the Associate became the IT Manager and she was appointed an assistant. As the project evolved, it became clear that packages would be a way of freeing up the Research Manager, IT Manager and the assistant. The key area this would help with was the reduction in problems arising from the Apple Mac infrastructure. For the Research Manager, the OneSource application was also explicitly linked with the ability of the package to be used by others and a lack of the need for her to develop the package further, thereby freeing her to

undertake more value adding work. It remains to be seen how the IT Manager's role will be impacted post 'go live' of the client-tracking project.

Consequently, it is not guaranteed that packaged software will 'totally' release information systems staff from development and maintenance activities. Yet, also if custom development does occur alongside packages (or indeed on its own), then this may contracted out anyway as demonstrated at the home wares company described by Light (2001a) and the web development project outlined by Levina (2002). Thus, if the need for custom development is recognised during or after the process of selection, questions are raised as to whether the reasoning behind the initial decision to implement packaged software might change.

6.2.8 Implementing Change to Working Conditions

A good deal of the literature on packaged software highlights that it may be used as an excuse to enact change because it embodies particular ways of work that the organisation should adopt in order to maintain the future support of the vendor (Champy, 1997; Glass, 1998; Soliman and Youssef, 1998). Therefore, this may be a hidden reason underpinning other arguments for adoption. At T.Co however, although there was an agenda of change, this was not initially linked with the adoption of packaged software. It was only as the packaged software was selected, implemented and used that this became a reason for its adoption.

The migration, OneSource and client-tracking projects brought with them explicit agendas for making the approach of staff at T.Co toward information systems more professional and streamlined. This was manifest in the various data cleansing

exercises and the implementation of common data entry standards throughout the organisation. Much of this activity was desired anyway, but it was later held as crucial due to the implementation of the packages. T.Co also had a history of monitoring of various kinds. This is evident in the Research Log application which was in existence solely to provide performance management information to the Operations Manager. Indeed, knowing at what stage of the process clients were at was a key part of the everyday life at T.Co. However, with the packages, T.Co's senior management wanted to increase the capability for monitoring of various forms.

As a result of the migration project, T.Co implemented a shared diary system using Outlook. This instantly allowed anyone in the organisation to see what anyone else was, reportedly, doing. Interestingly, the culture of the organisation was so used to monitoring, that staff expected this and accepted the application without any reported resistance (I was never able to find out if staff developed 'work arounds'). I did ask several members of staff how they felt about this and they were surprised that I would mind it myself. This answered my question!

In contrast, resistance was evident with the onset of the client-tracking project. Obviously, the name suggests that some form of monitoring was taking place. The case highlights how the consultants had a different agenda to the senior management of T.Co. In 1999, a senior consultant had reported that an application, such as the client-tracking package, would be very useful for monitoring client progress and also for helping them identify how much of their time they were spending with a client (and therefore how much money they were making or losing). By January 2001,

when a consultants' requirements meeting was held, the consultants' wish list questioned why a computerised tracking system was needed, for whose benefit was the system, what was its primary aim and how much extra work would it cause them. The consultant spokesperson at the meeting was worried that it was them, as well as clients, that were going to be tracked. This resulted in manoeuvring by the non-executive director present to repackage the project as one that would allow the consultants to make sure T.Co were keeping their side of the bargain. However, before the selection process, packaged software was never mentioned as a catalyst for change in respect of the desire to monitor. Indeed, whether senior management intended or not, the client-tracking package will offer capabilities to monitor process that it supports and by default, those that perform those processes. However, once selection had taken place, the IT Manager did begin to use the package to suit her own agenda. She claimed: "in preparation for the Goldmine project, our existing data will have to be cleansed and constraints put in place to prevent data becoming 'dirty' again". However, as the package is yet to be implemented, the extent of the changes to the working conditions, most interestingly in relation to the consultants, remain to be seen.

6.2.9 Bravado

There was definitely a good deal of bravado associated with the adoption of the packages in this study. The whole ethos of the projects during the time that I had spent with the company was grounded in bullishness and the desire to impress, aspects brought out in the Fashions and Fads Literature – cf. (Abrahamson, 1991; Newell et al., 2001). Although not particularly widely discussed in general in information systems, ideas related to this can be seen in the push of ideas such as

ERP and Knowledge Management (Galliers and Newell, 2001), BPR (Hammer and Champy, 1994) and in the light of recent events, such as the 'Dot.Com Boom' (Howcroft, 2001). At any opportunity, T.Co senior management would highlight their use of new technologies (the packages) as 'state of the art' and 'leading the way', although they would always temper this as being nothing without the personal service they provided. The company newsletter was the standard vehicle for this form of publicity. The Research Manager was quoted as saying "OneSource offers a competitive advantage in terms of quality of research and level of service to our clients. OneSource adds a professionalism and efficiency that should turn a Gold Service into a Platinum one." Also, on several occasions, the Managing Director, stated his personal objectives of the two year TCS as 'First Class Systems and First Class People'. The 'label' of the software was used to show off, impress and demonstrate legitimacy. However, the Managing Director also wanted to ensure that the system was "nothing without our staff" and therefore it was later re-labelled, when incorporated with the new TMP application, as the T.Co Research Database.

Much of this bravado was because T.Co wanted to capture a larger proportion of the market that was held by 'the big boys'. The assumption was that, with the new infrastructure in place, they would be able to 'take them on'. Moreover, because their clients mainly originated from 'Blue Chip' companies, they felt it necessary to prove themselves as a legitimate and suitable service provider. The bravado associated with the reasons for adoption was further used by various T.Co staff well into selection, implementation and usage. In addition to those examples above, bravado was present in determining which aspects of the functionality of a package T.Co would adopt, and which it would not, because their way was better.

The summary findings relating to the reasons for packaged software adoption, the purpose of this section, can be seen in Table 6.1 at the end of this chapter.

6.3 Package Software Selection: The Definition of User Requirements

The requirements gathering processes at T.Co have varied in length, depth and significance in relation to product identification, assessment, selection, implementation and usage.

For the migration related packages, the requirements gathering processes were minimal. In fact, as the process was viewed as a 'straight swap', the only implicit requirement was that they worked as least as well as the same packages had in the Apple Mac Environment. Even then this was assumed because of the 'reference' provided by the external IT Director and their usage by other organisations. The selection process was, in effect, not enacted.

For OneSource, the requirements gathering process was informal. A formal systems development methodology was not used in much the same way as is reported in some custom development projects (Fitzgerald et al., 2002). The purchase decision was base solely on the requirements of the Research Manager despite the expansion of the roll out to other areas of the organisation. It might be argued that the Research Manager was a very well informed user who knew T.Co business rules and requirements in great detail. Even so, other users, who may have made useful contributions, were ignored. However, even without the involvement of a number of users in the requirements gathering process, the application was initially received

very favourably. The view of the application has, however, changed somewhat following usage and consequent changes in objectives as discussed in section 6.4.1. Despite studies such as those by (Flynn and Davarpanah Jazi, 1998; Peppard, 2001), that state user involvement in the requirements gathering process for packaged software is incredibly important, it is arguable that this is not necessarily a prerequisite for success (Wilson and Howcroft, 2002).

In contrast to the migration and OneSource projects, the process of requirements gathering for the client-tracking project was intensive. Between August 1999 and December 2002, no less than six attempts at requirements gathering can be traced at T.Co. These involved many members of staff, although generally sourced from senior management. One attempt at requirements gathering was even initiated by Vendor E despite them previously being sent relatively clear documentation. In retrospect, it is possible to argue that the vendor wanted to use the package as the basis of the requirements gathering process in order to make their job easier and maintain the economies of scale on their part (cf. the appearance of 'absent' functionality when pressed by members of the project team at the workflow day).

Yet, ultimately none of the requirements gathering processes mattered as the Managing Director used his own internalised requirements to evaluate the package in the end. Furthermore, the extension of the coverage of the package at that time placed the requirements gathering process further into the hands of the Managing Director (and the vendor to some extent), especially as limited attention had been paid to obtaining the views of those in the area affected. Moreover, the vendor arguably took advantage of this to reorganise the implementation to suit their own

agenda. That is, to implement what they viewed as the most standard (and also the most financially rewarding) part of the project.

The ideal model of packaged software selection suggests that requirements gathering and user involvement are crucial to success. The data presented here however, highlight how requirements gathering may take on differing degrees of importance and also be enacted in more or less intensive ways. The case data also suggest that contrary to expectations, limited emphasis upon requirements gathering may not necessarily lead to failure and vice-versa (a point made in respect of more traditional custom developments by Wilson and Howcroft (2002). Finally, in a market oriented environment, it is clear that all requirements may not be perceived as equal, nor may they be understood up front. As Quintas (1994) states, different types of users (such as customers, managers, operatives) will have different degrees of access to developers at different times – what he calls 'forms of producer-user relations'.

At T.Co, the requirements were finally very broadly defined by the Managing Director in respect of Goldmine and the migration packages. When he was convinced that the packages could do what he thought they should do, progress was made. In a similar fashion, OneSource requirements were determined by the Research Manager, but the Managing Director had the final say. In each of these scenarios, other users, who were in some cases closer to knowing what their particular requirements may have been, were either not included in the selection process (OneSource and the migration packages) or were included only after the product had been selected (Goldmine). The effects of not involving other users earlier in the process of the Goldmine selection has further elongated the process of

requirements gathering beyond the selection of the product. Adding to Montazemi et al's (1996) study which suggests that information systems staff may lack organisational knowledge, it is possible to argue that various senior managers or end users may not have this knowledge either.

This may be a futile argument in any case. In respect of OneSource and the migration packages, it is interesting to see that a lack of user involvement has not led to total failure. It may therefore be that case that sometimes it does not matter whether users are involved and that in these cases it would be ethical for those in the position of determining this to explain this to users (which of course assumes they were interested in getting involved in the first place).

The summary findings of this section can be seen in Table 6.2 at the end of this chapter.

6.4 Packaged Software Selection: Evaluation

Various approaches to package evaluation are apparent in the T.Co cases. In respect of the migration project packages, these were instantly accepted as the commercial standard or the equivalent of what T.Co had. Therefore, there was no assessment of these applications and selection was automatic. There was minor wavering over whether Filemaker Pro would be replaced by Access but this was rapidly dismissed. The Managing Director felt that this would present too greater risk technologically and organisationally even though the financial cost of keeping Filemaker Pro was greater than that of moving to Access. In sum, evaluations were not performed to any great extent as there was no perceived alternative.

Similarly, OneSource was identified as the 'best' solution in a relatively simple fashion. The Research Manager had worked with it before, was viewed as the expert in the area and had the opportunity to sell the idea to the Managing Director. In this case it was also possible to trial the product as suggested by Kunda and Brooks (2000). This however, was a façade of an evaluation as it was not about determining if the package was the one to choose. It was about doing what was necessary in order to obtain the approval of the board. In this case it translated into using the software so the Research Manager could say, yes it does the job. Moreover, the Research Manager used this time to build a stronger business case for adoption by suggesting to the Board that the sales department could also benefit from its purchase.

The process in relation to client-tracking was much more complicated. Experimentation for example, was not a viable option. This was because the package would need to be implemented, widely diffused and widely infused in T.Co in order for any experimentation to take place. A variety of channels were used to identify suitable products though. As with OneSource, a member of staff, the Marketing Manager, had used a similar application in a previous role and interestingly the same application was chosen although it was never clear whether this was significant factor in its selection. The assessment process was fairly traditional in that a requirements document was created and sent out to selected companies for tender. Additionally, reference site and vendor premises visits were undertaken. The visits to vendor premises were used, in conjunction with mediocre sales presentations, to rule out products.

Initially, no explicit criteria were used to assess each vendor or the products they were selling. Instead, the requirements of each member of T.Co staff on the project team were interpreted as variously fulfilled on a presentation-by-presentation basis. It was only after three failed presentations that coaching was performed with a third vendor to link the package on offer with T.Co requirements. At the second presentation by this vendor, the Managing Director then made explicit his criteria by listing a range of business functions on a whiteboard and ticking these off as he asked the vendor if they could be met. At this stage, the explicit selection criteria were associated with functionality. However, the case also demonstrates softer criteria in terms of the mutual respect of the Managing Directors' of T.Co and Vendor E. The combination of a clear, personalised sales presentation and the development of a relationship with the client won Vendor E the contract for a product that had previously been dismissed.

In contrast to the ideal model, T.Co's experiences suggest that evaluation may not take place, and even when it does, criteria may be highly personal and politically loaded. To further explore these ideas, this section will now refer to the two most widely used evaluation criteria in packaged software selection – the functionality of the package and the vendors of the package.

6.4.1 Functionality Considerations

The functionality of packaged software is widely accepted to be a significant factor in the evaluation of packaged software (Lucas et al., 1988; Chau, 1994; Deloitte and Touche, 1996). However, much of the literature presents studies that suggest that people in organisations know what they want, or do not want, in order that they can

make the decision about whether packaged software, or a particular product, can satisfy their needs (Welke, 1981; Martin and McClure, 1983; Sharland, 1991; Nelson et al., 1996). In some ways at T.Co this was the case, and in others it was not, as is now demonstrated.

At the beginning of the project, the personnel of T.Co had varying levels of knowledge of what they wanted from the packages they were to implement. In respect of the Windows, Microsoft Office, Outlook and Sage applications, these were thought to be relatively clear as they had similar applications in their Apple Mac environment. Consequently no evaluation of functionality was performed.

In respect of OneSource, the requirements were also relatively simple and were easily satisfied initially. The Research Manager perceived that alternative packages were not available and that OneSource was the product of choice. Even in relation to the existing TMP application, there was no real decision to be made. The Research Manager knew what information was required by clients for conducting their job searches as this was very well established within T.Co. She also knew what OneSource was capable of due to her previous experience with the package. The OneSource experience highlights how, even where there is the potential for a decision to be made it may not happen if it is not perceived as such. Also, it is shown again that no evaluation of a product may occur at the time it is being selected.

The client-tracking project was another matter however. It was initially intended that Goldmine would be rolled out in the research department but this changed at the

successful sales presentation when the Managing Director 'saw' the capabilities of the package. The scope and budget of the project were instantly increased as the objectives for the package became more demanding. With respect to requirements, many members of T.Co staff were involved in various requirements gathering exercises in the run-up to the investment decision, and beyond. These began in 1999, and due to the effort put into these activities, it would have been expected that the requirements would have been fairly well defined. Indeed, the requirements documentation was very comprehensive. However, on many occasions the people involved were heavily influenced by existing ways of work, despite what they said about wanting to adopt best practices. The members of T.Co had been inculcated with the ways of work that were embedded in the Filemaker Pro applications. Unfortunately, the technical capabilities of this development environment (it was incapable of being used to develop relational databases, for example) and the lack of formal system development expertise, had led to very inefficient ways of work. The members of T.Co did not realise this, as it was all that they had ever known, but they knew that they were very busy and something needed to change. However, what they did not know, for example, was that mail merges could be done automatically, without selecting each individual record in a database manually, and that it was the data that were exported to a word document, rather than the word document being exported into the Filemaker Pro application. At meetings a frequent comment was that 'Filemaker Pro will not let us work like that'. As Gremillion (1982) points out, a package may not approach work in the same way a user would.

It really was not until various members of staff began to see demonstrations of various packaged software applications that they realised what was possible. This

was especially true for the Managing Director who, up to the personalised sales presentation, was more interested in going down the customised route. The problem then shifted the other way and the IT Manager was engaged in a process of managing expectations as the various groups increased their objectives for the package. The IT Manager was very successful in this. However, the picture changed again at the workflow day. The consultant attempted to further pare down expectations without finding out what the current expectations were and with little appreciation of the needs of T.Co. His attempts at driving through a standard blueprint of the package failed as those in the meeting withdrew from the process and viewed the package as incapable of supporting their needs. In the end, the Managing Director repackaged the project using his interpretation of the requirements. However, within these parameters, the Research Manager in particular is still pursuing the issue of functionality through the second workflow document and it is expected by the IT Manager that this will be further shaped by other staff members in the future before it is eventually signed off.

These experiences further suggest that evaluation of functionality was only undertaken where it was perceived by someone to be necessary.

6.4.2 Issues Related to Vendors

T.Co dealt with various intermediaries in the packaged software supply chain throughout the period of the study.

In order to implement the Windows, Office, Outlook and Sage Packages, T.Co sought the support of a vendor. Although there was the façade of a selection

process, the decision as to which vendor would get the job had been made very early on based upon recommendation by an external consultant. It would have been interesting to see if the decision would have changed if the other assessed vendors had performed better. Nevertheless, the IT Manager believes that the selected company have proved to be a good choice. The evaluation therefore was based primarily upon the recommendation of the external consultant and not an extensive checklist as suggested by Martin and McClure (1983) and Welke (1981).

The evaluation process for the vendors of OneSource was similarly very simple. Only the developing organisation sold and supported the product and therefore no evaluation process took place. On the surface, vendor considerations for the OneSource product became defunct as there was no choice as to who would sell or support the product. However, implicit within the case is the idea of whether or not the Research Manager would have chosen the product had her past experiences with it been problematic. I questioned the Research Manager about this and she said that prior support for OneSource had been good and she saw no reason why it would not continue to be in the future. In respect of evaluation, this implies that historical experiences of the package were transferred to the one in question as a short cut to evaluation.

T.Co used a charitable organisation to identify potential vendors of the client-tracking package. The result was that the T.Co IT Manager was bombarded with telephone calls and marketing information from many companies. From this, she identified four companies that T.Co would send their requirements to. Vendor D was the producer as well as the seller of the software and they instantly discounted

themselves as being too expensive for T.Co. In essence, T.Co were evaluated in contrast to the ideal model which assumes control by the consumer organisation. The other direct approach was to Vendor A, but they were discounted by T.Co as they seemed comparatively expensive and communications had been difficult from the outset. Vendors B and C, were the ones that were invited to give sales presentations – both were resellers rather than the developers of the products on offer. Vendor C was discounted merely as the product was not perceived as providing the functionality that T.Co required even though the seller was deemed satisfactory. Vendor B was discounted because the product (Goldmine), was not perceived as meeting the needs of T.Co. This vendor did not pay any attention to the needs of T.Co and demonstrated a standard, rather than personalised, product. In these cases, it was important that the vendor and the product stood up well to T.Co's requirements.

These results led the Managing Director to demand investigation into the capabilities of Filemaker Pro, and as part of this process, the developers of the package were approached and asked if it could satisfy T.Co requirements (of course they could as it was merely an environment within which custom development could be performed). Concurrently, the project team had continued their search for a package and had been approached by another vendor. An initial sales presentation was held to 'vet' the vendor without the Managing Director being present. The team knew that they could not afford another problematic presentation so this seemed to them to be the safest option. The strategy worked as they used the first presentation to further guide the vendor in relation to their requirements and this resulted in a successful sales pitch at the next presentation where the Managing Director was

present. This emphasises the diversity in, and political nature of, the package evaluation process, a point made widely in the custom development literature (Wilson and Howcroft, 2000).

Relationships with Vendor E initially started out well as the Managing Director of the Vendor E struck up a rapport with the Managing Director of T.Co. The IT Manager claimed that this had been a significant factor in the Managing Director's evaluation of the product. Indeed, the Managing Director was incredibly vocal in his praise for the approach of Vendor E because their Managing Director had been involved very early in the project. He saw this as "intelligent, respectful and the basis of a good working relationship". Therefore, it can be seen how very personal criteria can be used in the evaluation process.

However, when the consultant arrived for the workflow day he was very much geared towards offering a standard solution and had not appreciated the requirements of T.Co whereas those involved in the sales presentation had understood these. A similar situation arose in the study by Skok and Legge (2001). However, in contrast, there was no fear on the part of those at T.Co in respect of the consultant holding too much power, influence and knowledge that might be perceived as disappearing when he did. For example, Vendor E's consultant had obtained minimal information about the requirements of T.Co and this meant that he lost the respect of staff involved in the workflow day as he could not personalise the benefits of the package and realise his apparent agenda for implementing the package with minimal configuration. This demonstrates how evaluation continues after the 'evaluation period' in the ideal model. Goldmine had effectively been chosen at this point, but because of the

problems of the workflow day, based predominantly upon the skills level of the consultant, the project nearly halted before the final purchase was made. This further suggests that product selection does not automatically lead to purchase.

The skill sets of vendors are generally discussed in the literature in terms of whether or not these are available due to market conditions. The quality of the skills sets are also related to this, in relation to the idea that if demand for a package is high, then it may be difficult to obtain highly experienced consultants and implementing organisations may have to work with newly appointed consultants (Martin, 1998; Bingi et al., 1999; Sumner, 2000). In the T.Co case, the consultant was a long standing member of staff at the vendor organisation and this therefore adds further questions about the quality of services that a company may be buying to assist with packaged software implementation. As with custom development, it demonstrates an eagerness to instantly view the 'developer' as the expert. Moreover, it also makes explicit that multiple evaluations of vendors may be present, because they are not homogeneous entities and because they may be evaluated by various people, at various times and in a variety of contexts.

The summary findings of this section can be seen in Table 6.3 at the end of this chapter.

6.5 Package Software Selection: The Final Selection Decisions

In every purchase scenario, the decision to purchase was made by the Managing Director with reference to the Board. In contrast to the guidance of Kunda and Brooks (2000), the IT Manager was not the sole driver of the process, nor the one

who made the selection decision. The situation was in fact similar to Siemens Power corporation (Hirt and Swanson, 1999), where the information systems staff generated the conditions within which senior management could make a decision. At T.Co, they sometimes did not even do this. The study also reflects the view that the focus is on senior management rather than end user involvement (Sawyer, 2001).

In respect of the migration project, it was perceived by the Managing Director that there was no decision to make in respect of the packages. T.Co needed to adopt the industry standard. The only package decision that arose was concerned with whether or not the company should move their custom applications from the Filemaker Pro to the Access environment. Again, this decision was determined by the Managing Director who felt that staying with Filemaker Pro was best for the company. The real decision in respect of the migration project was related to who the implementation and support contract would be given to. Again, this decision was made by the Managing Director and, as mentioned earlier, although there was a façade of a selection process, personal recommendation rather than the implementation of an 'objective' selection and evaluation process was used.

It was also perceived that there were no alternative packages to OneSource. The Research Manager was perceived as an expert in the area, and she thought the application was suitable. A trial of the software was performed to demonstrate her argument and facilitate the development of a business case for presentation to the Managing Director and the Board. However, this was viewed as a matter of routine and really it was not the package that was being approved, it was the financial expenditure.

Despite an extensive process of requirements gathering and package evaluation, the selection of the client-tracking application, Goldmine, was based upon a sales pitch that convinced the Managing Director that this was the right product. To some extent, the coaching of the vendor by the project team relied upon the prior requirements gathering and evaluation processes in respect of client-tracking. However, the coaching had not taken account of the Managing Director's declaration that the package should be rolled out to other areas, where requirements gathering and evaluation had not happened. Moreover, even though the selection decision in respect of Goldmine has been made, purchase has yet to follow. Indeed, as mentioned earlier, the project nearly stalled after selection because of problems at the workflow day. Even after the recovery from this, the workflow document is still under negotiation and until then, there is still time for the project to be stopped and another product (or custom development) to be chosen instead. Indeed, the potential of open source software is currently being investigated as a mechanism for the development of a knowledge management application at the company.

A summary of the findings in relation to the final selection decision can be found at the end of this chapter in Table 6.4.

6.6 Implications of Packaged Software Selection for Implementation and Usage

This section briefly considers the issues from the cases that arise in respect of the implementation and usage of packaged software in relation to selection. It begins by examining some of the issues associated with implementation and then considers

issues of usage in terms of the functionality of the packages, vendors and implementation partners and customisation, maintenance and upgrade.

6.6.1 Implementation

Up to December 2002, only the packages associated with the migration and the OneSource projects have been fully implemented. Both of these implementations were perceived by the IT Manager and Managing director as being fairly unproblematic. The client-tracking project however, is only partially implemented and already has proven difficult. The projects are interesting from an implementation point of view as they provide further illumination on the potential links between this and selection. In order to discuss these potential links I refer to the broad categories relevant to success and failure that I introduced in section 3.6 on page 74.

It is argued that **user involvement** in selection is necessary in order to determine functionality requirements and facilitate change processes (Gremillion, 1982; Gibson et al., 1999; Al-Mudimigh et al., 2001; Akkermans and van Helden, 2002). However, user involvement was minimal in relation to the migration and OneSource projects yet broadly they were reported as being received satisfactorily. In contrast, in the light of the implementation activity that has been undertaken so far in respect of Goldmine, the consequences of user involvement are unpredictable. Certain users were involved in the selection process but the selection decision ignored much of this. As the early stages of implementation have been carried out, the users have become involved again and raised objections to the package. However, it remains to be seen whether, when the package is finally implemented, this will matter.

An appropriate business process change is often associated with successful package implementation and therefore it is argued that this needs to be considered in the selection process in terms of choosing a product that most closely resembles the intended ways of working (Lucas et al., 1988; Holland et al., 1999a; Markus and Tanis, 2000). In the migration project, the intended ways of working were not fundamentally different to those that were in place already. The changes that did occur related to the implementation of an electronic diary – which went smoothly, and the introduction of electronically supported remote working – which did not. However, even though the latter change had been anticipated and trained for, it had nothing to do with a misfit between the intended ways of working and the package. Rather, it was the ways of remote working made available by the package that were the problem. This suggests that changes associated with packages cannot be predicted with any certainty prior to, or in the process of selection, as it might not be until later that the full ramifications of their adoption are clear. This point is also made in two other packaged software studies (Hanseth and Braa, 1998; Hanseth et al., 2001)

Top management support in implementation is linked to selection on the basis that there are benefits to be attained from enrolling them early in the project. At T.Co, it would have been impossible not to involve senior management, partly because it was a small company and partly because of the organisational cultures. More accurately however, the T.Co experiences highlight that it may well be more important to have senior management feel as though they have been involved in selection. This is most starkly highlighted in the Research Manager's successful attempt to seek agreement from the Managing Director and the project teams engineering of the acquiescence

of the Managing Director to the purchase of Goldmine. This has significant consequences for the client-tracking project when problems arose because, as the Managing Director had sanctioned and taken credit for the decision, he had to intervene.

Implementation has been suggested to be more successful if **personnel capabilities** are sufficient (Holland et al., 1999a; Skok and Legge, 2001; Kraemmergaard and Rose, 2002). Therefore, the implication for selection is to ensure that the personnel capabilities, for the selected package, are present or obtainable. At T.Co, the personnel capabilities for the implementation of the packages were conceptualised as internal and external at the selection stage. The migration packages were implemented by an external vendor who was perceived to have the capabilities. The support of the IT Manager, an external consultant and the TCS academic team were also used. Indeed, the IT Manager was provided with extensive training that was tailored to the project at hand (in terms of familiarity with the setting up, and administration of, a Windows network). It was also identified that extensive training would be provided to the end users even though they had used the packages before in the Apple Mac environment. In addition, the Managing Director decided that to select Access over Filemaker Pro would be too risky as they did not have the skills in that area. For OneSource, personnel capabilities were not given much thought in the selection process as the Research Manager had used the application before and was also the Filemaker Pro developer in the organisation (so the link with the TMP application was not perceived as a problem). However, it was determined in implementation that assistance from one of the TCS academics would be required to help in making the link between the package and the TMP application. Aside from

this, the implementation of OneSource proceeded smoothly with a minimal amount of informal guidance to end users in terms of usage. Finally, in much the same way as the migration project, the personnel capabilities for the implementation of the client tracking project were thought of in terms of external and internal resources. Most significantly however, despite the vendor selection process eliminating those which were perceived as incapable, problems still arose into the beginning of implementation. Therefore, even if personnel capabilities are considered before implementation, there is no guarantee of success.

Understanding the capabilities of the package is assumed to be important prior to, and in, implementation because if this is the case, then the 'right' product is likely to be chosen and implemented leading to user satisfaction (Al-Mudimigh et al., 2001; Akkermans and van Helden, 2002). The experiences of T.Co suggest that understanding the capabilities of the package at any point is no guarantee to success. This is because it depends upon who's understanding of the capabilities emerges from the selection process. Even then, it ignores the point, as shown by OneSource, that the 'right' product, may become the 'wrong' product due to issues not associated with the technology in question. Moreover, even where the capabilities of the package may be universally agreed as appropriate, once implemented, they may bring unanticipated changes in working practices (Robey and Boudreau, 1999). Therefore, it is necessary to think of the 'capabilities' of the package, not only in terms of the technological capabilities, but also their capacity for 'requiring' new capabilities of those in the organisation.

Package software implementation studies suggest that **appropriate decisions regarding customisation and/or configuration** are critical to success (Holland and Light, 2003). T.Co, enacted their selection processes with the intention that they would not undertake package customisation. Therefore, packages were dismissed that were perceived as too far removed from T.Co requirements. However, the problems T.Co have experienced in relation to the client-tracking project, post selection may lead to the requirement to customise the software (although the issue has yet to be raised).

Recognition of an organisation's **legacy information systems** is argued to be an important feature of implementation because knowing the 'as is' situation is argued to afford an appreciation of the extent of the required change (Lucas et al., 1988; Al-Mudimigh et al., 2001; Holland and Light, 2003). The link between selection and implementation on this front is shown in many ways throughout the T.Co case. Examples include the involvement of the T.Co consultants in the requirements gathering process for the client client-tracking project, and the overall recognition of the move from a number of isolated systems to one common system which was anticipated by the Managing Director, very early in selection, as a 'total culture shock'.

It is argued that undertaking a **sound assessment of user requirements** for the package is necessary in order to scope the project as changes to these during implementation may increase timescales and costs (Lucas et al., 1988; Gibson et al., 1999; Markus and Tanis, 2000). A sound assessment of needs is therefore, highlighted as important at the time of package selection. Yet, the experiences of

those at T.Co suggest that this is not always the case. Requirements gathering was undertaken to varying degrees with no causal link to success or failure evident as a result. Moreover, this assertion assumes that user requirements are known in advice and ignores, again as shown at T.Co, the fact that these may emerge later.

6.6.2 Management of Change and Expectations

Finally, it is suggested that successful implementation requires the management of change and expectations in respect of the package (Markus and Tanis, 2000; Akkermans and van Helden, 2002). The selection process therefore links with implementation via notions of user involvement and curtailing the various sales pitches that might be made. This proved very necessary in the client tracking and migration projects, but not so much in with the implementation of OneSource. Ironically, where attempts were made at change and expectations management, early in the projects, this is where the most difficulties occurred.

6.6.3 Functionality

In terms of implications for usage, the functionality of a package is seen as important for two reasons. First, the package needs to do what it was purchased for and therefore a package should be selected that ideally can be used 'out of the box'. Secondly, however, it may be that there is no package that possesses the functionality, out of the box that an organisation may require, and therefore, in order to attain appropriate functionality, customisation may need to be performed (this is discussed in greater detail in section 6.6.5). The purpose of this section is to focus upon the functionality issues arising from usage.

It was thought that the migration project involved a 'straight swap' of Apple Mac packages for Windows packages and, for the most part, this view was correct even though it brought a new feel to the familiar applications. However, for most of the staff the Outlook application was an entirely new package with which to work. Therefore, training for the use of Outlook was provided to all staff by a specialist training company - another intermediary. In general, the IT Manager felt that the training provided was adequate and this was further supported by the development and implementation of a super-user policy, that over time, helped with the settling in of the new portfolio. However, although the training helped these staff familiarise themselves with the canonical processes, it did nothing to assist with learning about new non-canonical processes (Lee and Lee, 2000). This resulted in remote staff being able to use Outlook without having learned that they did not need to save files in RTF format any longer as they were operating in a Windows environment, that RTF files took up more bandwidth than files with a .doc file extension and that they needed to synchronise their folders regularly. The system ground to a halt as the staff did not know how to work with it. They continued to use RTF files as they did not realise that this was no longer necessary. Also, as they had never had email or electronic remote access, they were used to working on a higher capacity internal network, which did not require them to log on and off via the secure gateway software. Now they were bound by the limitations of their home 56K maximum internet connection.

This problem was presented very firmly at the door of the vendor, the IT Manager and myself by the Managing Director because he said the 'technology was not working'. The training organisation was never approached. Clearly, the problem

did not just relate to technology. No party predicted that the problem would arise; no one had experienced this problem before. Initially, the vendor thought that it may have been that the lines used for the Virtual Private Network might have needed increased bandwidth. However, it was their analysis of the network traffic that highlighted the problems stemming from the characteristics of remote worker usage. Although the problem was re-conceptualised by the IT Manager, academic team and the vendor, none of the T.Co complainants ever modified their view of themselves at least in public. The problem remained one of the 'technology' rather than one brought about by particular usage of the technology. This point is speculative but it would have been interesting to see if the problem would still have arisen had a more formal requirements gathering process been undertaken which better accounted for the idiosyncrasies of the working practices and skills levels of those at T.Co.

Usage of OneSource was initially much more straightforward even though staff had not been involved in its selection and it had been rolled out to areas for which it had not previously been intended. Even in the light of this, the package was initially reported as being a great success by all who used it. However, because the package was opened up to a wider group than was originally intended (from Research to the Sales and Marketing function), towards the end of the study, these requirements have altered and the package has been highlighted as deficient. For the Sales and Marketing staff, what was the 'right' product, has become 'the wrong' one, even though they did not have an application like this before. It is impossible to determine if the requirements gathering process had been more formal, and had involved the Sales and Marketing staff, whether this would have changed the situation. Indeed, it appears that the success of the package, in terms of its

acceptance by the Sales and Marketing department, has led to more demanding objectives for the package which were not intended nor can be satisfied - that is, the inclusion of overseas company data and non-corporate organisations. This scenario is similar to that described by Adam and O'Doherty (2000) where, as managers learnt about the possibilities of a package, their objectives became more ambitious. In addition, it reinforces the work of Wilson and Howcroft (2002) who state that failure (and by implication success) may not be unilaterally agreed, nor hinge on the technical functioning of a system.

6.6.4 Vendors and Implementation Partners

It is far too early to draw any major conclusions from the selection process in respect of its long term effects vis-à-vis the choice of vendors and other implementation intermediaries. However, enough data are available to highlight that the use of 'objective' evaluation processes may or may not be used, carry weight and result in the expected outcomes.

Although the migration project involved a formal evaluation process, this really was circumvented by the Managing Director's highly valued criteria of personal recommendation which ensured that the preferred vendor was chosen. Despite the lack of 'rigour' in the process, to date, those at T.Co report that they have made the best choice. Similarly, in terms of the OneSource product, positive experiences have been reported by those who use the application, even though the vendor was not evaluated at the time of the purchase.

In contrast, despite the 'rigorous' evaluation process that was conducted in order to find a vendor and implementation partner for the client-tracking project, difficulties post selection have become apparent. The difficulties with the implementation partner are well documented in terms of the implementation partner's capabilities to deliver the product they sold to T.Co. Given the effort that went into this process in contrast to the migration and OneSource projects, this is surprising.

For each project, it remains to be seen how T.Co's experiences of the selected vendors and implementation partners take shape over time.

6.6.5 Customisation, Maintenance and Upgrades

T.Co have not customised the Windows, Office, Outlook and Sage packages as they did not need to do so. Indeed, it is arguable that most organisations would not need to customise these kinds of applications to any great extent with perhaps the exception of Sage.

Maintenance agreements were implemented for these packages but interestingly issues of lock-in and switching costs were not placed upon the agenda. Clearly, the migration to the Windows environment, and what were perceived as standard industry applications, minimised thinking about this. It was felt by those at T.Co that they were already locked-in, but to the wrong platform and associated applications. Indeed, one of the reasons cited for the move to the Windows environment was to increase the range of applications T.Co could access. Furthermore, as T.Co opted (whether knowingly or otherwise), to be locked into certain Microsoft products, it is possible to argue this was not a major issue as the

majority of other companies were in the same position and therefore although they were path dependent, they were in the major path.

In respect of OneSource, the package itself was not customised. However, new functionality was added in the form of the linkage with a revised TMP application. As Light (2001a) suggests, this may lead to the requirement for in-house maintenance if the OneSource package changes. Even if the new client-tracking package replaces the TMP application, the linkage will still be present and this increases the chances for further reworking of any linkage between the two as both developments leave the hands of T.Co.

The client-tracking project has not yet seen the requirement for customisation and it is here that it is likely to be the most costly. At the close of this study, T.Co appear to be happy to bend towards the package and not customise the product. However, it remains to be seen, in the light of the rejection of two workflow documents, as to whether this viewpoint is maintained.

In relation to maintenance and upgrades, the consequences of the selection processes enacted for each of the packages remains to be seen. A summary of the findings for implementation and usage can be seen in Table 6.5, at the end of this chapter.

6.7 Summary and Conclusion

This chapter has presented an analysis of the field study data with reference to the conceptual framework that was developed from the literature review. The aim has been to demonstrate the problems of existing theories of packaged software selection

and provide a basis for the development of an alternative theory. In order to more easily identify the bases for addressing the two aims of the chapter, I have summarised the findings in Table 6.1 to Table 6.5. Each table is a summary of findings in relation to the various constructs in the conceptual framework. The findings suggest that the process of packaged software needs to be theorized in a different fashion and this has resulted in revisions to the conceptual framework that are discussed in the next chapter.

Table 6.1: Summary Findings - Reasons for Packaged Software Adoption

Reasons for Packaged Software Adoption

- T.Co wanted to standardise to facilitate working with others and also to buy into non-value adding standardised practices. T.Co's practices were perceived as 'best' practices. However, the adoption of the technology was sold to customers as T.Co having best practices.

- Despite packages being 'sold' as the replacement for legacy information systems. T.Co's did however operate custom and packaged software therefore packages can also be viewed as legacy information systems.

- T.Co were sold packaged software products

- T.Co were sold the idea of packaged software over custom development

- The costs of packaged software were given as a reason to abandon custom development and also stay with it.

- The costs associated with packaged software are demonstrated as potentially being more difficult to predict than reported in the literature.

- T.Co was more interested in buying into a support network for a 'tried and tested' product rather than a rigorously built product. T.Co was never really interested in the mechanisms by which a packaged software product had been built and whether it worked well for other purchasers. Even when they undertook reference site visits, they did so with the intention of evaluating support mechanisms rather than products as they felt they operated completely differently to those that they visited.

- The case for freeing up information systems related staff was used a few times, as a reason for packaged software purchase in terms of justifying investment on economic and strategic grounds

- Packaged software was explicitly chosen to facilitate the implementation of senior management agendas for change Some agendas were made widely known and others were not. These agendas, public or 'outed', were subject to varying levels of resistance.

- Bravado was used on many occasions to fuel decisions to adopt packaged software

Table 6.2: Summary Findings – Definition of User Requirements

Definition of User Requirements

- The requirements gathering processes varied in length, depth and significance in relation to the evaluation and final selection decision processes
- Varying degrees of involvement of end users in the requirements gathering processes was evident despite guidance and reports in the literature
- Limited emphasis upon requirements gathering, and involvement of end users, may not lead to failure and vice versa
- Reasons for packaged software adoption (and non-adoption) may emerge from the requirements gathering process
- Requirements may be determined by the vendor or implementation partner

Table 6.3: Summary Findings – Package Evaluation

Package Evaluation

- Package evaluation may be conducted in a more or less intensive fashion and this may not impact upon the perceived success of failure of the process
- Evaluation can be a difficult process as it is not always the case that requirements are known or can be known. In addition, requirements may change as evaluation proceeds
- Evaluation criteria are dynamic, highly personal and may be variously interpreted as rational/irrational
- Perceptions and feelings may be used in evaluation in favour of 'hard' criteria
- Some requirements may be ignored when it comes to evaluation, in favour of the agenda/s of specific individuals or groups
- The evaluation process may also lead to requirements gathering
- The packaged software consumer organisation may be evaluated by a vendor or other implementation intermediary
- Reasons for and against packaged software adoption may emerge from the process of evaluation

Table 6.4: Summary Findings – Final Selection Decision

Final Selection Decision

- The purchase decision may or may not be based upon all or part of the requirements gathering and evaluation processes
- The reasons for the final selection decision may be simultaneously interpreted as rational and irrational
- The purchase decision is highly political
- Making the decision to purchase does not mean that a purchase will be made

Table 6.5: Summary Findings – Implementation and Usage Issues

Implementation and Usage Issues

- Requirements gathering may continue
- Evaluation may continue
- The 'right' product may become the 'wrong' one
- Lock in wasn't a barrier to selection and was perceived in some ways as useful as it afforded stability in the business environment.
- The selection process has implications for implementation and usage

7 Summary and Conclusion

7.1 Introduction

This chapter summarises the findings and presents the conclusions of the study. It begins with a review of the research objective and an evaluation of the methodology. This is proceeded by the presentation of the overall conclusions of the study leading to a revised theoretical framework. Recommendations for future research are the offered and the study is drawn to a close.

7.2 Review of the Research Objective

Despite trends towards the implementation of packaged software in work organisations (and society), the information systems research community still primarily focuses upon developing custom development theories, concepts, methodologies and the like. Moreover, even where packaged software is the focus of study, most of the attention is directed towards the nature of the application type. Whilst useful, this approach is somewhat limited in terms of contributing to the development of concepts and theories about packaged software. Consequently, the

research in this thesis aims to raise the profile of packaged software as an area worthy of serious attention.

More specifically, this research has questioned the utility of existing theories of packaged software selection. It is argued that packaged software selection theories rely upon a unitary view of work organisations which ignore the presence and importance of differing organisational realities. Moreover, it has been demonstrated that existing theories primarily attribute control of the selection process to the consumer organisation when clearly market forces may influence this in different ways. Additionally, is suggested that, in reality, the selection process may not play out in the well ordered and linear way it is currently presented.

The objective of this study was, therefore, to develop an alternative theory of packaged software selection. The intention was to develop a theory that would more plausibly account for the multiple realities of the process and the consequent potential 'disruptions' to what is usually described as a well ordered, linear process. In order to realise this objective:

- Chapters 2 and 3 review the existing research on packaged software with particular emphasis upon processes of selection. Chapter 2 examines the market-oriented nature of packaged software and chapter 3, the organisational perspective. The underlying thought of each chapter was how the issues that emerged may relate to the process of packaged software selection. A conceptual framework, incorporating an ideal model of packaged software selection, which maps out the landscape of the literature in relation to packaged software selection, is developed on the basis of these two chapters.

- Chapters 5 and 6 tell the story of organisational experiences with a variety of packaged software products and emphasise the problems of the 'ideal model' of packaged software selection reported in the literature. The conceptual framework developed earlier is used to structure this analysis. The analysis leads to the development of a revised theoretical framework that is presented later in this chapter.

7.3 Evaluation of the Research

In chapter 4, a set of criteria to guide the conduct and evaluation of the research were formulated. These criteria were: contribution to knowledge, appropriate methodological choice, the fundamental principle of the hermeneutic circle, the principle of contextualisation, the principle of interaction between the researchers and the subjects, the principle of abstraction and generalisation, the principle of dialogical reasoning and the principle of multiple interpretations. In this section, the research is evaluated on this basis. Moreover, it is important to note that the evaluation explicitly incorporates an assessment of the strengths and limitations of the research.

7.3.1 Contribution to Knowledge

Myers (1997) argues that a good indicator of interpretive research is that it offers a contribution to the information systems community in respect of whether the author has developed new concepts, applied new concepts or applied a well known theory in a new or unique way. He also suggests that a good marker is research that offers rich insights into the human, social and organisational aspects of information systems development and application or if it contradicts conventional wisdom.

This study makes a contribution in the following ways:

- It (further) develops the concept of packaged software
- Chapters 2 and 3 offer an extensive review of the packaged software literature which offers rich insights into existing research in the area related to human, social and organisational aspects of information systems development.
- The experiences of T.Co offer rich insights into human, social and organisational aspects of information systems development particularly in relation to packaged software (although parallels and implications are also drawn for custom development too).
- The findings contradict the rationalistic theories of packaged software selection.
- An alternative theory of packaged software selection is put forward.

One limitation of the contribution is that it could be argued that the findings are based upon an insufficiently diverse data. For instance, it is suggested that theory building from case data usually requires approximately 4-10 cases (Eisenhardt, 1989). Therefore, on the surface, as the study is based upon one organisation's experiences with packaged software selection, this could be argued to be problematic. However, as stated in chapter 4, case studies should be defined by an interest in individual cases (Stake, 1994). In this study, there are many individual cases of packaged software selection, because multiple perspectives of the three projects were actively sought and presented in the data. Even so, there is still the question of how representative the experiences at T.Co might be. It could for instance be suggested that much of what happened was because it was occurring in and around T.Co and that other organisations might provide significantly different

insights. Indeed, this is certainly the case. However, it is arguable that the way that the data have been used to develop the alternative theory has lead to findings that are sufficiently abstract to have broader appeal, yet utility. Nevertheless, it would be interesting to extend the study to incorporate the experiences of other organisations and different packages.

7.3.2 Appropriate Methodological Choices

There is consensus that methodological choices should be commensurate with the research objectives (Lawler, 1985; Silverman, 2001). Therefore, qualitative research was undertaken in order to develop a better understanding of how packaged software selection is enacted and why it might occur in certain ways. Clearly, a survey for example, would not have provided the rich insights into the process of selection that were necessary in order, as is recommended, to study over time a phenomenon in its real life context (Yin, 1989).

Initially, it was intended to conduct case study research, but as the research progressed, the opportunity presented itself to obtain very rich data, with relatively secure access, via action research. Indeed other aspects of the methodology emerged in a far from deliberate fashion as realistic researchers suggest is the case (Pettigrew, 1985; Paul, 2002). However, although the methodology emerged, it was also rigourous. Great care was taken to follow the guidelines of case study research and then action research. Data collection and theory development was also monitored and refined as necessary.

Further potential limitations of the methodology, and the ways in which attempts were made to deal with these are discussed as the research is further evaluated according to the principles of Klein and Myers, as detailed in sections (7.3.3 to 7.3.8).

7.3.3 The Fundamental Principle of the Hermeneutic Circle

This principle upon which the following ones expand, suggests that we come to understand 'the whole' by considering its parts and their interrelationships. The 'whole' in this study refers to packaged software selection. Consequently, it is possible to develop a better understanding of packaged software selection through the parts and interrelationships it comprises. Thus, to understand packaged software selection more thoroughly, the constituents of the packaged software selection and their interrelationships were initially drawn from the literature and presented in the form of a conceptual framework which was used to guide the field work. The field work further focussed upon gathering data about the constituents of the selection processes and their interrelationships.

It is, however, important to note that it is incredibly difficult to understand the totality of a packaged software selection process in theory and practice. The problem therefore, was determining when there was 'a whole' that was useful and thus when data collection and theory development could stop. Data collection was stopped when it became clear that nothing of great significance was being added to as a result of additional data – that is, when theoretical saturation had been reached (Eisenhardt, 1989; Strauss and Corbin, 1994).

7.3.4 Contextualisation

The principle of contextualisation requires that the subject matter be situated in its contexts so that the emergent setting under investigation can be communicated to the audience. For this study, I was fortunate to have been involved with T.Co for some time before much of the activity in respect of packaged software selection took place. I joined the organisation in June 2000 and work did not begin properly until November that year. This time also allowed time for acclimatisation in the organisation, especially as the decision to undertake action research was made fairly early on. This action research approach also afforded regular access to the company that has allowed for quite significant insights into the contexts of the packaged software selection processes.

Such good access was no guarantee to an 'accurate' portrayal of the contexts for the selection processes however. Even where I was heavily engaged with the organisation, there was no guarantee that I would find out about the 'real' data about situations such as conflicting viewpoints. Therefore, in order to improve the accuracy of the reporting of the context, various sources were used to clear up any misunderstandings and to try to find out as much about how situations were viewed by those involved, whether this was through documentation, cross checking formal/informal interviews or observation. This process also aided the explicit consideration of my role, and that of others, in the development of the research as discussed next.

7.3.5 Interaction Between the Researchers and the Subjects

This principle requires the researcher to explicitly consider their role and that of others in the development of the research. The action research strategy used in this study clearly implies that there should be interaction between the researcher and the subjects in developing the outcomes. The evidence for this is in the case itself where my role as action researcher 'advisor' to T.Co is outlined as is the role of others in making decisions and acting on these in everyday life.

Thus, in respect of the field study, the aim was for data collection, reporting and analysis that was as faithful as possible to the various viewpoints of those involved. This was achieved through the application of the meta principle of the hermeneutic circle and then by cross checking my interpretations with other members of the academic TCS team and those others involved in the packaged software selection processes. The aim was to develop shared meanings of the data in question at any given time. The findings of this study therefore comprise interpretations of my own and others who were involved in the study. However, as different people interpret data in different ways, this leads to multiple realities (Kaplan and Duchon, 1988). The study aims to communicate, and heavily relies upon, these diverse realities in order to development the alternative theory of packaged software selection. However, because I have 'controlled' the study, the work is ultimately presented from my perspective, a typical criticism of interpretive studies.

7.3.6 Abstraction and Generalisation

Abstraction and generalisation are not prerequisites for interpretive studies. However, this study does offer opportunities for relating ideas to multiple situations

although it is important to point out that these are not characterised as universal laws as with positivistic studies (Denzin, 2001). Instead, this research offers generalisations that relate to the development of concepts and theory which may be intersubjectively understood to be useful (Walsham, 1995b). These generalisations relate to the further development of the concept of packaged software and theory about packaged software selection. The contribution in this respect was outlined in section 7.3.1 and is discussed in greater detail in section 7.4.

7.3.7 The Principle of Dialogical Reasoning

This principle advocates sensitivity to the possibility that theoretical preconceptions might influence the findings of a study. First, it is suggested that the researcher should make the philosophical assumptions of the research as transparent as possible and relate the strengths and weaknesses of this to the work at hand. In chapter 4, the philosophical (interpretivist) underpinnings of this study are clearly articulated and critiqued. Moreover, the difficulties of the approach are alluded to in this evaluation section (for example in relation to the idea that it is the researcher who has the final say over what gets interpreted - written down - and how. This principle also suggests that the research findings might not support the initial theoretical preconceptions of the study and that the researcher must be aware of the need to revise these as necessary. There are five key points at which the theoretical preconceptions were altered as a result of the study. Illustrative examples are provided in Appendix B.

1. This study was very much influenced by two years' previous experience of undertaking research into ERP packages in a business school setting. As

mentioned in chapter 4, the original study was intended to be about another application type (CRM). This was the source of the first contradiction between the very loose theoretical proposition (i.e. the theory of CRM software) and the shift towards a study packaged software. It was during the initial literature review that my position shifted as a 'new story', albeit from the literature, began to emerge. That was, packaged software selection represented a potentially more relevant and theoretically rewarding area of investigation in information systems.

2. This shift in position led to a review of the various theories of packaged software selection and from this emerged the ideal model (Figure 3.1). In conjunction with the fieldwork for the migration and OneSource projects, it became clear that existing theories of packaged software selection were not necessarily played out in organisational settings in a well ordered fashion. The ideal model of selection gleaned from the literature was therefore adapted in the light of this (see iteration 1, Appendix B).

3. The revised model was further questioned when the client-tracking project field data began to emerge and the roles of the 'market' started to become clearer. This resulted in further theoretical revision (see iteration 2, Appendix B). This was later further affirmed by aspects of the OneSource project and, as a result, the migration project was revisited too.

4. As the OneSource package bedded in, and the client-tracking project selection process further progressed, the origins of the beginning and end of the selection

process were questioned, leading to another cycle of revision (see iteration 3, Appendix B).

5. Finally, as the study was written up and the theoretical preconceptions were further refined, a final conceptual framework resulted. This is presented later in this chapter.

This principle compares sympathetically with the theory-action cycles required of action research.

7.3.8 The Principle of Multiple Interpretations

The principle of multiple interpretations requires the researcher to examine the influences of social context upon actions in the study. This means seeking out multiple viewpoints and the reasons for these. The whole basis of this thesis is the idea that there are multiple realities that may influence any given packaged software selection process making this perhaps one of the most important principles in this study. Consequently, a broad variety of viewpoints have been made evident in the literature review, but more particularly in the fieldwork. For example, the T.Co case illustrates differing viewpoints between the following individuals and groups:

- Consultants/T.Co senior management;
- T.Co senior management/T.Co staff;
- T.Co Managing Director/T.Co IT Manager;
- Vendor E/T.Co end users;
- T.Co senior management/myself.

Where these differences in perspective have arisen, the aim has been to find out about, and tell the story of why this may have been, or was perceived to have been the case.

7.4 Interpretation of Research Findings

The findings from the experiences of T.Co in relation to packaged software selection are summarised in the concluding section of chapter 6 and therefore I do not intend to repeat these in any detail here. In this section, however, I wish to draw out the main overall findings from the research. This includes the development of the concept of packaged software and, the primary contribution, an alternative theory of packaged software selection. The theory is represented in a theoretical framework and is compared with the conceptual framework, presented in Figure 3.2, that was used to guide the research.

7.4.1 The Development of the Concept of Packaged Software

Typical definitions imply that packaged software is pre-built, licensed for use and commercial in nature. In this thesis, there is a development of this concept in recognition of the idea that some packaged software is often 'given away' – web browsers and Internet email application such as Alta Vista for example. Consequently, it is suggested that packaged software should be thought of as 'pre-built software that is intended to be licensed for use'. This concept has scope for packages that are licensed on a no charge basis. The concept is also inclusive of the variety of package applications types available including the commonly researched ones that are used to support organisational activities and those that have received less attention such as operating platforms and those found in wider society. Those in

the latter group include operating platforms, entertainment applications such as PC/Console games, reference works such as Microsoft Encarta and 'Domestic Automation' packages such as those to assist in home improvement, legal and financial management activities.

7.4.2 An Alternative Theory of Packaged Software Selection

The major contribution of this study has been to develop an alternative theory of packaged software selection. In chapter 3, it was shown that existing theories of packaged software selection broadly comprised a linear model of activities associated with identifying user needs, evaluating software on the basis of those needs and then selecting the most suitable packaged on this basis. Drawing upon the literature and undertaking the study of T.Co brings out a range of issues that suggest the process is far from certain, linear and objective in nature, as summarised in Table 6.1 to 6.5 of the previous chapter. These findings form the basis of the revisions to the conceptual framework resulting in a refined theoretical framework that represents a development of the theory of packaged software selection. The revised framework is illustrated in Figure 7.1.

Whilst this framework may look like it is an attempt to develop an objective embodiment of reality in respect of packaged software selection, I would draw on Lewis' (1994) perspective of entity models which have been subject to the same label. Lewis (1994) states that entity models can also be viewed as a representation of an interpretation of reality, and therefore, may be used as a way of negotiating a shared understanding of a problem domain. Thus, the framework above can be used to view various perspectives of the packaged software selection process in order to

facilitate an appreciation of multiple realities rather than the reification of 'one, true reality'.

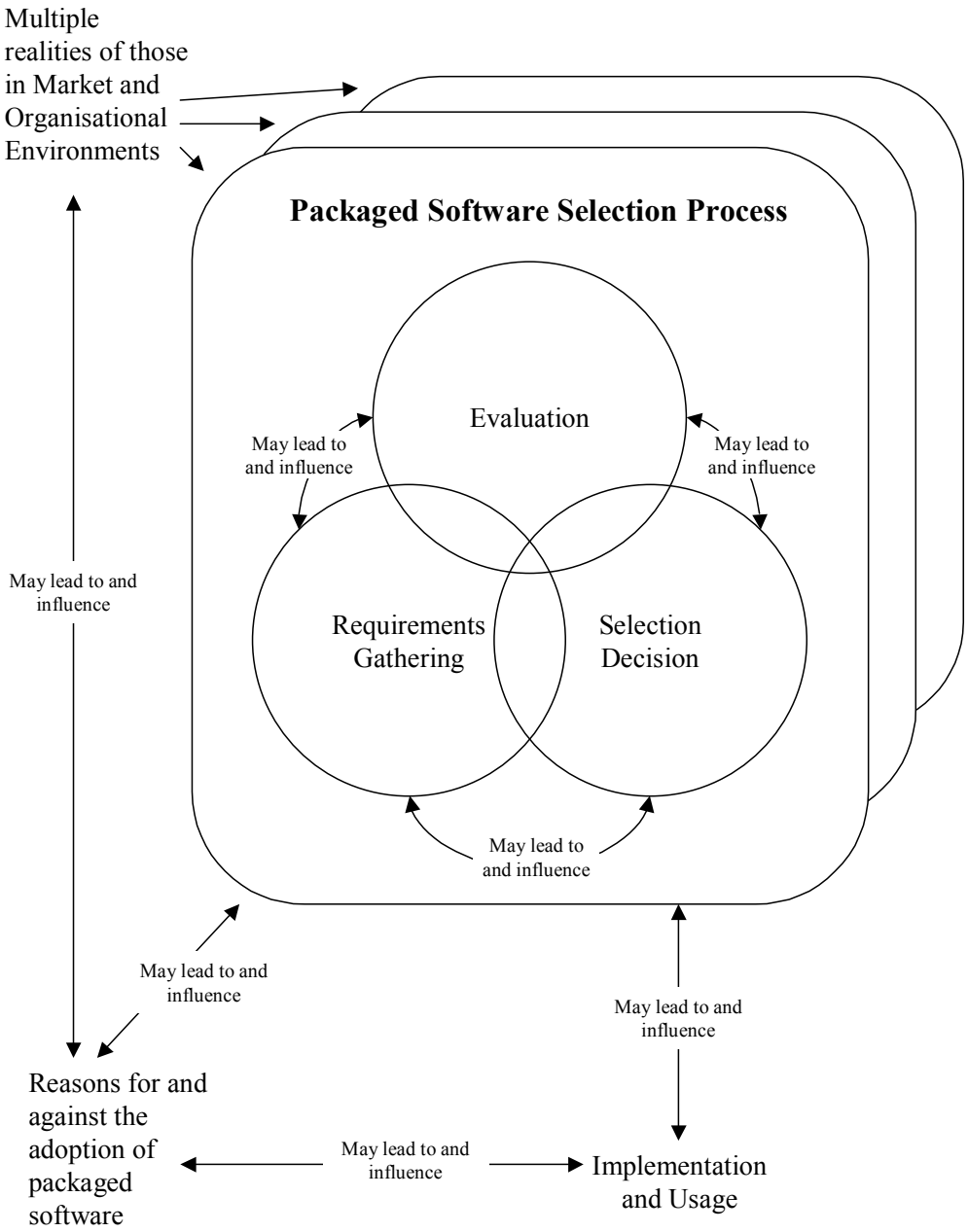


Figure 7.1: The Revised Theoretical Framework

In order to explain the alternative theory in greater detail, the constituents of the revised framework are now discussed.

The framework explicitly acknowledges the presence of ***multiple realities*** of the packaged software selection process. Essentially, this refers to the idea that various groups or individuals will be involved in the process either as part of the organisation environment into which any selected package may be implemented or from the market environment where the package was created. Because of these multiple realities, it is possible that differing views of what is rational may emerge. Furthermore, these multiple realities may bring with them multiple agendas which may complicate the process of selection and may mean it is not enacted in a rationalistic fashion in the way that the initial conceptual framework, and more specifically, existing packaged software selection theories may suggest.

The presence of multiple realities may shape ideas about ***reasons for and against packaged software adoption*** before, during and after the selection process is enacted. This is in contrast to the initial framework as there is no assumption that those involved in the selection are aware of the reasons why they may or may not choose packaged software before they embark upon the process. The framework suggests that these reasons can emerge before the decision to adopt packaged software is made in the same way as the initial framework. However, as the selection process is enacted, it is argued that further reasons why packages should, or should not, be adopted may emerge. For example, they may emerge through requirements gathering or evaluation activities. Similarly, reasons may also emerge when a package is implemented and/or used in an organisation. The emergence of these reasons, at whatever point, may lead to the continuance or discontinuance of adopting packaged software or indeed a particular product.

Again, in contrast to the initial framework, the ***implementation and usage*** of a package that is subject to the selection process is not guaranteed. This is because the selection process itself is not treated as having to be completed. However, where a package is selected, implemented and used, the nature of the selection process can impact upon this. For example, as with custom development, minimal user involvement in selection may or may not hinder implementation and subsequent user acceptance. Furthermore, it is useful to note that the process of selection continues into implementation and usage. As people get more involved with a package, they may change their perceptions of that package, and reasons for or against may emerge as requirements gathering and evaluation processes continue. Both of these may result in decisions to continue with the package or not – even sometime after the package has 'gone live'.

In terms of the heart of the framework, the selection process, ***requirements gathering*** is still included. However, the revised framework clearly acknowledges that differing, potentially competing sets of requirements will be brought to bare in the process of selection. Critically, these requirements may be determined by those in 'the market' as well as those in the selecting organisation. Further, because of the nature of packaged software development, requirements may be indirectly dictated by others in different organisational and market environments. Questions therefore arise as to whose requirements are used, if any, in the selection process. Although requirements gathering is included in this framework, it has been shown that whether this is played out or not depends very much upon the nature of the selection process and the product in question. Finally, the logic of the framework asserts that 'all' requirements may not be known at the end of any requirements gathering process

which is different to the initial framework where these appear to be set before evaluation takes place. This is because processes of evaluation and selection decisions resulting in implementation and usage may reveal new requirements or quite simply they may change.

The ***evaluation process*** also remains in the revised framework. However, although the results of the evaluation process may be used in making the final decision about which package is to be implemented, there is no guarantee of this. Similarly, there is no guarantee that any evaluation may occur in the first place. Furthermore, as with the rest of the process, evaluation is not viewed as a one-time effort. Evaluation may occur numerous times throughout the process of selection, and beyond, in order to obtain new requirements or inform a selection decision. Moreover, because the framework emphasises the role of multiple realities, there are consequently multiple, co-existing, and sometimes competing, evaluations that further complicate the process of selection.

Finally, the ***selection decision*** is included. In the initial framework selection is argued to be undertaken on the basis of the evaluation process. However, it is clear that this may not necessarily be the case. Decisions may be made on the basis of no evaluation and the effects of this may be variously interpreted as positive and negative. Moreover, if the selection decision is a 'no', this may result in the need for further requirements gathering and evaluation. This is similarly the case where the decision is a 'yes', because the product enters the implementation phase and usually results in usage. This is for two reasons. First, the selection decision does not necessarily imply that a purchase will be made – at T.Co for example,

implementation began even though a contract had not been signed. The contract still has not been signed, and will not be, until the requirements have been finalised to T.Co's satisfaction. Therefore, the selection decision does not automatically mean that implementation will begin, or will be completed. Secondly, even if a package is purchased, and implemented, it may not be used, or used as widely as expected.

In summary, the revised theory suggests that existing theories of packaged software selection are flawed and naive conceptualisations of the realities of the process. Whilst the concepts of requirements gathering, evaluation and the selection decision are useful to help us understand the process of selection, it has to be pointed out that they cannot be viewed or arranged in a simplistic fashion. Realistically, attempting to simplify packaged software selection to some extent is useful and necessary (in fact this is what the alternative theory here does) however, this study demonstrates the problems of over-simplifying the process to the point where it becomes meaningless and inaccurate.

7.5 Recommendations for Further Research

The study presented here, and specifically the theory of packaged software selection put forward, suggests several avenues for further research. Of course it is impossible to draw out every possible aspect worthy of further investigation, so instead listed below are a few that I think are particularly interesting, relevant and significant:

Packaged software and custom software have been viewed as different in terms for example of development (Sawyer, 2000) and development/implementation (Kelly et al., 1999b). However, this study suggests, that there are also many similarities

between the two approaches. Specifically, custom development could benefit from conceptualisation from a market-oriented perspective. Although it is suggested that custom development is undertaken internally within an organisation (Krishnan, 1998), the use of external contractors is widespread (Light, 2001a; Levina, 2002). Further research might also look more closely at the parallels between custom or packaged software development and implementation in organisations in terms of the rhetoric of user involvement and agendas for information systems-driven organisational change.

The processes of selection described in the study indicate political and power implications for the adoption of packaged software due to the existence of multiple realities. Other studies have suggested that, like selection, product development is not a clear-cut or rational process of innovation diffusion and adoption. Rather products are gradually shaped through cyclic processes of versions where inclusion is selective (Scott and Kaindl, 2000; Light, 2001a; Pozzebon, 2001). Although politics and power have been studied in relation to custom development, it would be interesting to see the similarities and differences emerging from packaged software contexts especially where levels of commercialisation and supply chain intermediation are high.

The market-oriented perspective on information systems development suggests that consumer organisations need to rethink their approach to the information systems function in terms of skill mix and roles (Sawyer, 2001). Perhaps the most important theme relates to the competence of non-information systems and information systems personnel to undertake the necessary organisational and information systems

related activities that the selection of packaged software requires in a similar fashion to other sourcing agreements (Currie and Willcocks, 1998; Willcocks and Sykes, 2000). Further research in this area would therefore be useful.

There is scope for investigating the role of fashions and fads in information systems in relation the academic and practitioner communities. Theories of fashion setting networks suggest that innovations diffuse in terms of social band wagon effects where information about adoption and the adopters create social pressures for further adoption rather than the innovation per se (Newell et al., 2001). In the context of information systems, and specifically packaged software, this further emphasises the problems of rationalistic theories of selection. For example at T.Co. a great deal of bravado was used to justify selection in order for the company to retain legitimacy in it's field rather than improve efficiency. This idea has already been examined somewhat in respect of the spurt of ERP studies that were witnessed in the late 1990s (Westrup, 2002). Yet, there is further scope for other contributions.

Although the ethics of software vendors have not been a raised to any great extent by this study, the revised theory presented here suggests it is a matter worthy of further investigation. For example, the sales pitches of companies such as SAP used in selection processes, have widely reported the package as providing enterprise-wide support for business processes, yet organisations, such as the University of Nebraska, have found fit between the package and their processes to be as low as 30 per cent in some areas (Sieber et al., 2000). Further into the lifecycle, it would also be interesting to study the effects of lock-in and monopolistic positions such as that held by Microsoft in the market for domestic computing operating systems.

Finally, further research into the issues surrounding the use of packaged software in domestic settings is required given the increasing pervasiveness of home computing - in particular, the use of marketing approaches and how these might influence selection processes in contrast to those played out in organisations. The theory presented in this study would certainly have resonance with domestic settings, but it is an area that requires further investigation.

7.6 Conclusion

This study demonstrates that packaged software selection is far from objective and certain. The multiple realities associated with packaged software selection play out in various ways beginning with the decision to adopt packaged software over custom development and through to the context of usage. This study therefore asserts that whilst it may be necessary to simplify approaches to packaged software selection in order to make progress, over simplification can be problematical.

The wider implication of the study is that it draws comparisons between packaged software and custom development. Packaged software research has a chequered history, often being cited as a completely different approach to software development and implementation. The problem with this assertion is that the lessons about custom development have not been carried over to packaged software research as appropriate. This has, ironically, resulted in attempts to 'custom build' the wheel when one could have been 'bought' through reference to custom development (and potentially vice versa). Indeed, one of the underlying threads of this thesis is a basic lesson from the traditional systems development lifecycle. That is, be aware that mistakes made later in the process will usually cost a lot to undo.

Appendix A: Legacy Information Systems (Working Paper)

AN ALTERNATIVE THEORY OF LEGACY INFORMATION SYSTEMS[29]

Ben Light, Information Systems Research Institute, University of Salford, UK.
Pre-publication draft, not for citation.

Abstract

Organisational dependence upon IT continues to grow yet experiences of satisfaction vary widely. Problematic, aged IT is often cited as being a fundamental problem in this respect and this is commonly termed legacy information systems. However, in this paper the author offers an alternative, and arguably more comprehensive, theory of legacy information systems that accommodates multiple viewpoints and recognises its inherent dynamism. The paper suggests a theory of legacy information systems that comprises of the concepts of temporal effects, interpretations and characteristics. It is argued that legacy information systems are constructed of many 'legacies' that are handed down continuously, forming an amorphous set of socio-technical interdependencies and relationships.

Introduction

Organisational dependence upon information technology (IT) continues to grow and is at the heart of many organisations operations and strategies, particularly in the 'e-era'. In the midst of this, it is easy to focus upon the latest technologies, organisational concepts or just the 'buss' of the moment and hope to reap instant rewards (Markus and Benjamin, 1997; Howcroft, 2001) - but what about fundamental issues such as legacy information systems that we do not traditionally view as interesting? The benefits of, information systems are espoused by vendors, consultants, people in organisations and academics yet we still struggle with them.

[29] To be presented at the 2003 European Conference on Information Systems, Naples, Italy.

Legacy information technology, information systems and legacy systems have become widely favoured terms for the description of problematic IT by such groups. Although many studies have argued for multiple perspectives in areas such as systems thinking (Checkland, 1981), systems development (Wood-Harper et al., 1985) and strategic information systems (Ciborra and Associates, 2000), attention to legacy information systems is scant. In response, this paper explores the multiple meanings and characteristics of legacy information systems. I have adopted the term 'legacy information systems' for the remainder of the paper as it this is more inclusive and comprehensive than legacy IT or legacy Systems[30]. I aim to demonstrate the connections and overlaps that exist amongst the constituents of legacy information systems and recognise that attempts to demarcate the subject introduces problems. In offering a structure for the argument the idea is that the overall flavour of legacy information systems emerges.

In the next section, I outline an alternative theory of legacy information systems. This is followed by a detailed review of the characteristics of legacy information systems as reported in the literature – the review is organised in subsections dealing with issues of longevity, functionality, perceptions of technology and structure. The review is used to exemplify the proposed theory and is followed by a summary incorporating an illustrative theoretical framework.

Multiple Views of an 'Old' Problem

Throughout the field of IS the concept of legacy information systems is widely recognised. However, the most pervasive notion has been that of technology that is

[30] Legacy IT focuses upon technology and legacy Systems ignores the informing role that the term legacy information systems captures.

old, outdated, in a state of disrepair and that does not satisfy organisational demands (Nassif and Mitchusson, 1993; Alderson and Shah, 1999). Furthermore, they are commonly associated with high maintenance costs, obsolescence, poor documentation and lack of technical support (Warren, 1999). As Wendy Robson notes:

Whilst legacy information systems a beneficial thing in lay terms in an IS management sense it represents the problems caused by *past* acquisitions that no longer fully match *current* needs. (Robson, 1997: 456)

Of course, the list could be extended, yet it communicates the idea of the dominant perception of legacy information systems as problematic (and we shall explore the range of characteristics in greater depth as the paper progresses). I would suggest that legacy information systems are not necessarily problematic or focused on IT. Instead, legacy information systems can also be viewed as an asset to an organisation. It has for example, been argued that the real value to the organisation of legacy information systems lies in the accumulation of business rules, policies, expertise and know-how that they embody (Kim, 1997). However, I do acknowledge that problems can arise when the legacy information system inhibits organisational adaptation to environmental change or strategic vision (Alderson and Shah, 1999). Although legacy information systems are often linked with IT, it is important to highlight the inextricable interdependencies and relationships with the context in which it resides. This has been referred to as organisation/business legacy and is, to some extent, a demarcation of the IT and Organisational context (Kawalek and Leonard, 1996; Bryant, 1998; Kelly et al., 1999a). Considering the

'softer' aspects of legacy information systems is essential as will be highlighted throughout this paper.

Strictly speaking, 'legacy' does not embody problematic connotations, nor does it necessarily embody good things. Legacy merely implies something that is handed down and is therefore, in some senses, neutral (Allen, 1990). Therefore, legacy *information systems* should be viewed as handed down IS. We can argue however, that the concept legacy information systems embodies less than neutral characteristics when situated in a context. In-context legacy information systems may hold favourable and unfavourable properties. These properties may also be subject to interpretation via different lenses. For example, the process of implementation of call centre technology into a previously paper based sales department leads to a revision in legacy information systems conditions. At the same time this introduces processes that allow senior management to monitor telesales staff performance. We can see that the legacy information systems conditions improve for senior management, yet they may deteriorate for those that are being monitored more closely.

Furthermore, legacy information systems are not static. Robson (1997) in her discussion of systems migration states that a legacy information systems issue is concerned with how to keep enough processing and human resources working to permit the phasing out and replacement with new systems. Laudon and Laudon (2000) offer insights in much the same vein indicating that, typically an organisation's legacy does not support newly designed business processes and that these business processes are islands in a sea of inherited legacy business practices from long ago. Singh (1997) also refers to legacy as the 'previously installed'. The

term installed, implies that 'once it's in, it's in' and it will not change. Legacy information systems are dynamic, particularly when the in-context situation is an organisation. In the previous example of the call centre, the IT based replacement for the paper-based system merely becomes part of the legacy information system. The process of implementation could also be seen as part of this. If the senior management communicated that the technology was being implemented to monitor staff for instance, this would undoubtedly influence staff perception of the modified legacy information systems.

Towards a More Comprehensive View of legacy information systems

This section reviews the literature relating to the dominant characteristics traditionally associated with legacy information systems in order in order to highlight the problems of the current theories. The issues are categorised as being associated with longevity, functionality, perceptions of technology, and structure. The author is aware that these issues are rooted in IT and that the argument for the theory of legacy information systems recognises organisational and social issues. Admittedly, much of what follows is very much the traditional view of legacy information systems as bad, old technology. The review merely reflects the literature, even though on the surface it may be seen as conflicting with the aims of the paper. In response, the author prefers to see the review as a useful platform for the development of an alternative theory of legacy information systems by highlighting the futility of predominantly technically loaded thinking.

Longevity

Henderson (2000) rightly points out that as systems get older, inter-relationships evolve resulting in complex dependencies amongst various system components. Lauder and Kent (2000) infer a problematic consequence of the longevity of legacy information systems stating that mature IS grow old disgracefully. They argue that although their lives may begin with a flexible architecture, repeated waves of hacking [changes] tend to petrify them resulting in accidental, inflexible architectures. Adolph (1996) concurs stating that constant patching makes the system's reliability questionable. Swanson and Beath (1989) also argue that as a system grows, it loses architectural integrity, staff become less familiar with it and maintenance activity increases. Therefore we have issues of complexity and inflexibility.

The Swanson and Beath (1989) view introduces further issues related to staff. These include the need to consider the capability of the organisation to be able to maintain and develop legacy information systems that has been in existence for a number of years. This issue can then also be divided into two areas: capability to do it for themselves or availability of someone else to do it for them. Legacy information systems may well have been constructed by people with skill sets that are scarce or unavailable in the current organisation and its environment (Singh, 1997; Taylor et al., 1997). In 1995 for example, it was stated that fewer people were available with skills in the programming language, Assembly (Bennett, 1994). In the late 1990s there was a deficit in certain packaged software configuration skills (Sumner, 2000; Willcocks and Sykes, 2000). These skills and knowledge sets therefore become incredibly valuable to user organisations and contractor/vendor organisations. This

situation may also be exacerbated when organisations are pushing and being pushed to develop IS staff to equip themselves with the skills sets necessary to implement new and emerging technologies (Brancheau, Jans et al. 1996). There are also issues related to this in terms of perceptions of legacy information systems. Are they just seen as what Adolph (1996) terms an 'old piece of crap'? Obviously the skills problem, discussed above, could also be applied to external sources, however there is a further problem from the external perspective in terms of the availability of the hardware and software itself (Adolph, 1996). How do people in organisations deal with discontinued product lines if a piece of equipment fails or they need to re-install software they have lost? The author encountered an extreme example of this when engaged on another piece of research in 1998. An external consultant working for a large retail group mentioned in an interview that he had spent the last four days searching the United Kingdom for a Winchester disk (which he obtained in the end!).

Functionality

Issues of functionality manifest themselves in many ways. The most obvious is concerned with whether functionality meets existing requirements and, where it does not, how easy this is to remedy. As Ramage et al. (2000) describe, legacy information systems is often what is left after a change has occurred in the organisation, but not the system(s), leading to a gap between the functionality they provide and the needs of the organisation. Again, we can see connections amongst the characteristics of legacy information systems as contributors to this state of affairs. For example, in the accompanying covering letter of the Kearney Management Consultants (1984) report it is suggested that it may be related to changes in emphasis from the 1960's and 1970's where relatively inflexible, costly

and complex machinery was used to automate simple repetitive tasks. Looking at this in greater detail, Adolph (1996) suggests that functionality difficulties may occur for instance where users want features that can't be grafted onto the 'old' system, where there are limited storage capabilities or unusual restrictions such as 80 column records (perhaps due to a short term view of the longevity of the system, or limitations of the build later). The issue of functionality, where this becomes problematic, is to some extent concerned with responsiveness Edwards et al. (2000). Changes in functionality may be required to respond to changes in the organisations macro-environment. However, legacy information systems may also be responsive to user needs. Bennett (1994) points out that the system may still do useful work and be reliable. However, he also notes that users may extensively rely on undocumented features meaning that although the system may have value, issues may arise if that functionality fails at any time. Recently, work has emerged which attempts to elucidate the problems of evolutionary complexity. Kaasbøll (1997) focuses upon developing an understanding of adaptive maintenance in relation to how numerous improvements may add up to negative effects. In his example, several user led functionality improvements to the legacy information systems gradually slowed down the operation of the system to such an extent that the users complained. Kaasbøll (1997) cages this as ‘counter finality’ - the concept that individual actions may create collective results that were not intended.

Perceptions of the Technology

Many people in organisations do not realistically anticipate how long the information systems that they introduce are going to last for. For example, in 2002, one of the largest financial services companies in the UK is still using legacy information

systems that were developed in the 1960s using the Assembly language. Indeed, the year 2000 revealed many problems inextricably linked with perceptions of how long the various information systems would be in use for. The problem with not perceiving that technologies may be in use for a number of years leads to the emergence several of issues. In consideration of the dynamic characteristics of legacy information systems it has been conceptualised as forming part of the environment for new developments (Warboys et al., 2000). As reported several years ago:

> "difficulties are foreseen in interfacing with existing equipment or systems. To a large extent, this is a criticism of the suppliers of both office automation and data processing facilities who have failed to anticipate future needs and supply flexible interfaces. It is also a result of a piecemeal approach to finding a solution where short term expediency has been taken at the expense of long term integration. Today's opportunity to buy a multiplicity of different equipment was quoted as 'leading to greater, and not fewer, such problems'."
> (Kearney Management Consultants, 1984: 27)

Even though the quote was taken from a report dated 1984, organisations are still experiencing legacy information systems problems bound up with this issue. It is clear that that people in organisations and suppliers, need to take a longer-term view of what is essentially the development of their legacy information systems. The problem also involves academia and society in general. For example, associated with the idea of longevity are peoples' attitudes to how long they anticipate working with an information system and this has no doubt been influenced by a number of groups. Lehman (1980) for instance, argues that in the 1950's the ecstasy of instructing a machine to undertake computations at undreamt of speeds overshadowed the rather dull need for a guiding theory and discipline. Equally, it has also been suggested that developers think 'legacy information systems' work is

tedious (Adolph, 1996), and that they may prefer to work on new developments instead (Bennett, 1994). How many undergraduate and postgraduate students want to learn about Fortran or Cobol after the advent of Java, Visual Basic and XML? Moreover, it has been suggested that, historically, IS development has received a higher priority than maintenance due to the growth of IS in the 1960's and 1970's (Swanson and Beath, 1989). This may lead to action not being taken that attempts to synchronise the software with the organisational needs, the absence of knowledge management regarding legacy information systems and consequently, the need for the deployment of increased levels of resources required to understand legacy information systems when maintenance does occur (Bennett, 1994; Taylor et al., 1997).

A further issue associated with perception is organisational comfort in dealing with legacy information systems problems. It is suggested that a dominant characteristic of problematic legacy information systems is that they are confusing, not well understood and display high levels of entropy (Taylor et al., 1997; Kelly et al., 1999a) even though developers may assume that organisations know all about them (Adolph, 1996). This confusion is equally traceable with regard to perceptions of what may 'solve' the problem. Legacy information systems is often cited as embodying the critical business processes. Indeed, Ramage et al. (2000) suggest that legacy information systems may be the sole explicit embodiment of a business process that has otherwise passed into tacit knowledge. Lauder and Kent (2000) concur in arguing that legacy information systems can be an asset, as through years of debugging effort, they will have grown to reflect essential tacit knowledge. A different approach is taken by Alderson and Liu (2000). When trying to understand legacy information systems, they assume that any documentation, including source

code, is out of date and inaccurate. They also assume that staff with technical knowledge of the system are unavailable. Their suggested approach is to observe the operationalisation of the system. Another view suggests that due to enormous investments in software, there is little likelihood that it can be replaced, it would be too costly to reprogram and there is minimal knowledge about what it does. Therefore, despite the imperfections of legacy information systems [where they exist] portions must be reused (Blum, 1996). There are problems with all of these approaches: i) assuming that legacy information systems reflect reality denies the existence of users developing coping strategies such as 'work-arounds', ii) relying solely on observation of the operationalisation of the legacy information system cannot afford a full appreciation of the detail of its workings and iii) to assume that in *every* situation that it is not possible (or necessary!) to rebuild legacy information systems runs counter to organisational experience - not least the trend toward the replacement of bespoke/custom software with packaged software.

Structure

It has already been suggested that, over time, the structure of legacy information systems may degrade and this can lead to problems in understanding how the system works for the purposes of development and maintenance activities in response to organisational requirements. Bennett (1994) suggests that legacy information systems will usually be characterised by absence of structure, little documentary support and generally only source code as a lens for understanding. He also suggests that the reason for this state of affairs can be linked to the early days of software engineering where system clarity and structure were subservient to the need for operational speed. Even though state-of-the-art techniques may have been used at

the time, such an approach may encourage maintenance that rapidly degrades structure. The views of Lehman (1980) and Arthur (1988) agree with this arguing that software changes may have been poorly designed, imposed and implemented with little thought given to the need for a possible re-write of the design or the potential for the introduction of bugs. Kearney Management Consultants (1984) have also stated that short-term expediency has overtaken the need for a view of long-term integration. Moreover, the problem is exacerbated by the lack of formal documentation. If documentation does exist, there is often a discrepancy between the documented description of the system's function and its actual function (Adolph, 1996). The problem of having to deal with untidy structure is compounded in consideration of the dimensions of the legacy information systems. It is argued that large software programs will comprise thousands of lines of code. Arguably this makes them difficult to deal with, unlike small software programs (Bennett, 1994). Unfortunately, the issue is not that simple. The complexity of the legacy information systems cannot be viewed in 'lines of code' terms. Even from a technical perspective, the complexity may be in the line of code its self rather than the number of lines. How does the line of code interact with other lines of code? What language has been used? How common is it? Who understands it? Is the program connected to others? Critically from the organisational perspective we need to consider the importance of the program and to whom.

Discussion

The analysis of literature reviewed for this paper reveals the prevailing view of legacy information systems as old and problematic. However, the literature also opens up areas where the value of legacy information systems is very much subject

to interpretation. The fact that a system (and its constituent parts) has been in place for some time does not necessarily equate with negativity. People in organisations that are part of the legacy information systems will be valuable in themselves as may be the technical component - it may contain data of value and staff may be comfortable with its outputs and the way it operates. Fine-tuning over a number of years may also mean that even though it is costly to maintain, it may support distinctive competitive strategies. Conversely, more recent legacy information systems may display problematic characteristics for many reasons including the way they were introduced or the skills available to support a technical aspect if it has proved popular with other organisations (such as ERP packages).

Despite predominantly focusing upon the technical perspective of legacy information systems the literature opens up inextricable links with 'non technical' issues which offer valuable support for the argument for a more comprehensive and inclusive view of legacy information systems. A lack of technical expertise to maintain a system (new or old) is obviously not a technical problem although it is often treated as such. It is perhaps more readily though of as a staffing problem. Similarly, whether the technical component of a particular legacy information system is viewed as problematic or favourable may very much be concerned with the perceptions of the role and longevity of it by organisational members.

Finally, contrary to their general treatment in the literature, legacy information systems can be interpreted as dynamic. The environment comprises technical, social and organisational artefacts and we can see that these are constantly changing and impacting upon each other. For example, the technical - through development and

maintenance activity, and the organisational - through staff turnover, and strategic responses to macro-environmental changes.

Figure 1 details the thinking presented in the paper in the form of a theoretical framework. It uses three concepts – characteristics, interpretations and temporal effects.

'Characteristics' refers to the nature of the legacy information systems. In this paper I have compartmentalised these under the headings of longevity, functionality, perceptions of technology and structure. Clearly, these are artificial boundaries and badges, as the characteristics of legacy will vary dependent upon the situation and interpretations of that situation as shown in figure 1. That is, for something to be characterised as legacy information systems, it has to be interpreted as such.

'Interpretations' are concerned with how the characteristics of legacy information systems are viewed – essentially this highlights its subjective and interpretive nature. The argument here is that different viewpoints of the characteristics that constitute the legacy information systems may exist. Note however, that some of these may have greater resonance due to the formations of intersubjectivities. What is not discussed here is which characteristics are the ones that are 'accepted' and worked with – that is beyond the scope of this paper but clearly various areas of interest such as information systems maturity and evaluation hold potential interest here.

'Temporal Effects' suggests that legacy information systems are subject to time-based considerations. Essentially, the characteristics of legacy information systems

will change over time due to such things as modifications to code, functionality extension and staff changes. As a result of these changes, and changes in context, the way that these characteristics are interpreted will change too.

In summary, the framework asserts that legacy information systems will have a variety of characteristics that are variously interpreted over time.

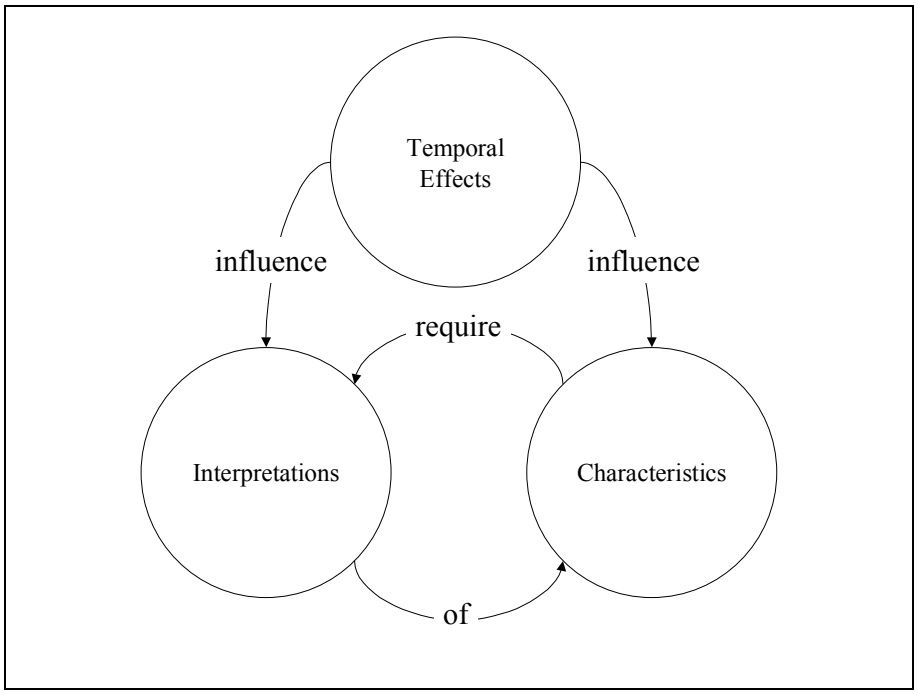


Figure 1: A Theoretical Framework of legacy information systems

Conclusions

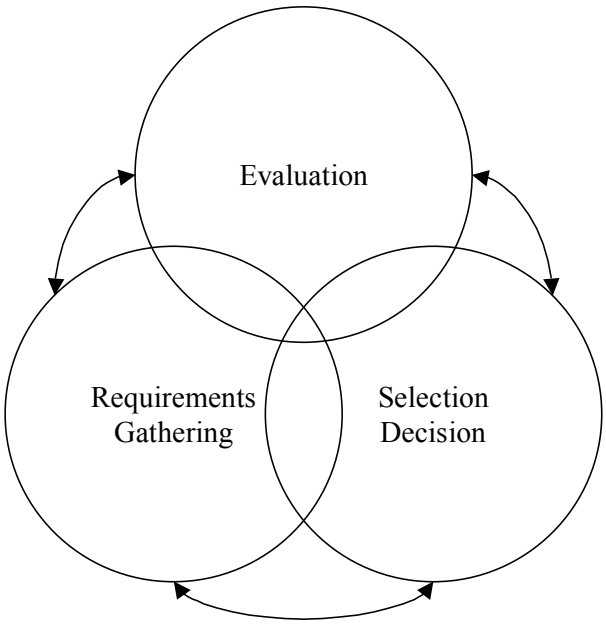
This paper acknowledges that organisational dependence upon IT continues to grow although many still struggle to obtain the full benefits from its implementation and usage. One source of the problem that is strongly represented in the literature has been conceptualized as legacy information systems. Unfortunately, the concept has been used illegitimately throughout the Information Systems and Computer Science academic communities, and the commercial world. Connotations of legacy information systems are generally strongly linked with ageing, static and problematic

IT although some aspects of the literature do argue otherwise. However, these arguments have not been translated into a comprehensive view. The alternative theory of legacy information systems that I offer here attempts such a view. I am acutely aware that there may be problems with existing information systems that form part of organisations, particularly where these have existed for some time. However, it is also clear that legacy information systems (old, and not so old), may sometimes offer immense value to organisations, particularly when the concept is broadened to include social, organisational and technical considerations. Perhaps therefore discussions should revolve around legacy information systems problems and legacy information systems benefits. A final important strand in the paper is the advocacy of legacy information systems as inherently dynamic. The interactions amongst social, organisational and technical factors are made clear – processes of implementation, maintenance and organisational restructuring are some of the exemplars that highlight this point very well. I have been involved in discussions regarding a system developing into a legacy information system on many occasions. Perhaps the answer in consideration of the alternative proposed here, is that everything already is a legacy information system.

Appendix B: Illustrative Examples of Theory Development

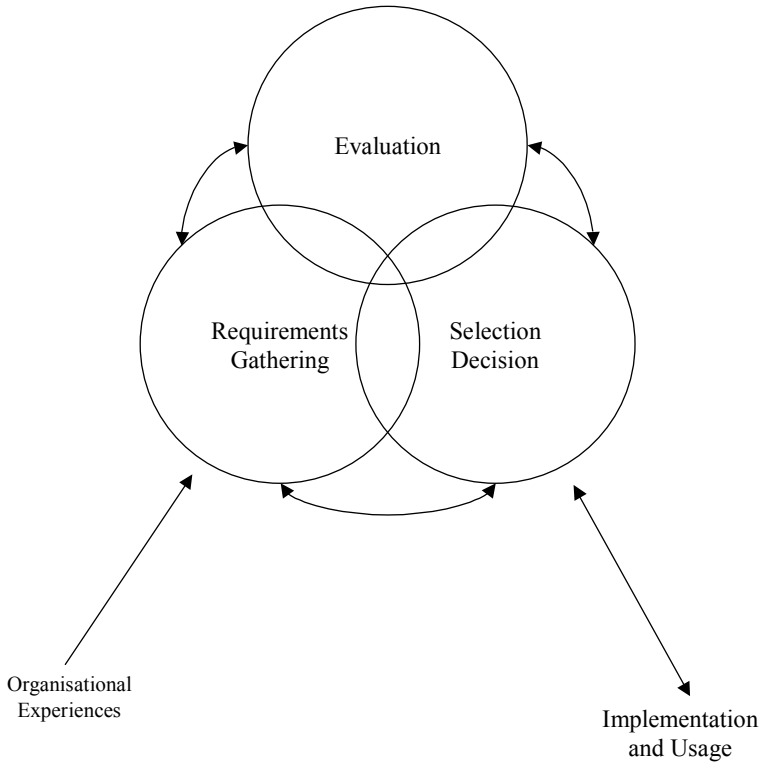
Iteration 1

Packaged Software Selection Process

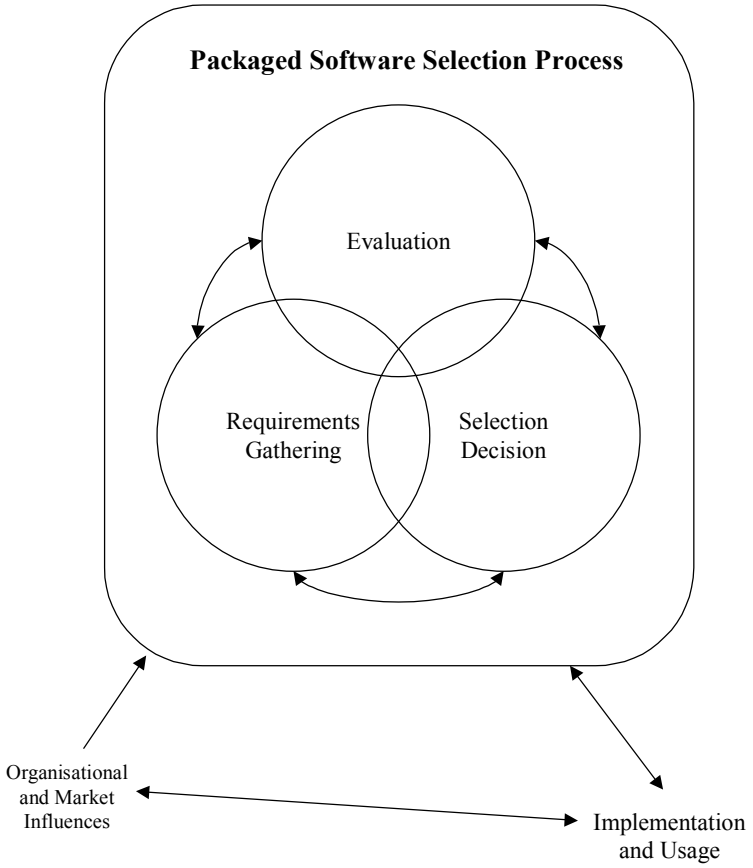


Iteration 2

Packaged Software Selection Process



Iteration 3


Packaged Software Selection Process
Evaluation
Requirements Gathering
Selection Decision
Organisational and Market Influences
Implementation and Usage

References

Abrahamson, E. (1991), “Managerial Fads and Fashions: the Diffusion and Rejection of Innovations”, *Academy of Management Review*, 16(3), pp. 586-612.

Adam, A. (2002), "Exploring the Gender Question in Critical Information Systems", *Journal of Information Technology,* 17(2), pp. 59-67.

Adam, F. and O'Doherty, P. (2000), "Lessons from Enterprise Resource Planning Implementations in Ireland - Towards Smaller and Shorter ERP Projects", *Journal of Information Technology,* 14(4), pp. 305-316.

Adolph, W. S. (1996), "Cash Cow in the Tar Pit: Reengineering a Legacy System", *IEEE Software,* 10(5), pp. 41-47.

Akkermans, H. and van Helden, K. (2002), "Vicious and Virtuous Cycles in ERP Implementation: A Case Study of Interrelations Between Critical Success Factors", *European Journal of Information Systems,* 11(1), pp. 35-46.

Alderson, A. and Liu, K. (2000), "Reverse Requirements Engineering: the AMBOLS Approach", in Henderson, P. (Ed.) *Systems Engineering for Business Process Change: Collected Papers from the EPSRC Research Programme*, Springer, London, pp. 196-208.

Alderson, A. and Shah, H. (1999), "Viewpoints on Legacy Systems", *Communications of the Association for Computing Machinery,* 42(3), pp. 115-116.

Allen, R. E. (1990), *The Concise Oxford Dictionary of Current English,* Oxford University Press, London.

Al-Mudimigh, A., Zairi, M. and Al-Mashari, M. (2001), "ERP Software Implementation: An Integrative Framework", *European Journal of Information Systems,* 10(4), pp. 216-226.

Anderson, E. E. (1990), "Choice Models for the Evaluation and Selection of Software Packages", *Journal of Management Information Systems,* 6(4), pp. 123-138.

Andersson, R. and Nilsson, A. G. (1996), "The Standard Application Package Market - An Industry in Transition?" in Lundeberg, M. and Sundgren, B. (Eds), *Advancing your Business: People and Information Systems in Concert*, EFI, Stockholm School of Economics, Stockholm, pp. 1-24.

Arthur, L. J. (1988), *Software Evolution: The Software Maintenance Challenge,* John Wiley and Sons, Inc, New York, NY.

Attewell, P. and Rule, J. (1984), "Computing and Organizations: What We Know and What We Don't Know", *Communications of the Association for Computing Machinery,* 27(12), pp. 1184-1201.

August, V. (1999a), "ERP Sites Hit by Performance Dip", *Information Week,* 17 February, p. 12.

August, V. (1999b), "Escape Modules", *Information Week,* 31 March, pp. 36-38.

August, V. (1999c), "Re-animator: ERP Special Report", *Information Week,* 28 July, pp. 23-32.

August, V. (1999d), "Up Close and Personal", *Information Week,* 28 April, pp. 4-5.

Avison, D. and Fitzgerald, G. (1995), *Information Systems Development: Methodologies, Techniques and Tools,* 2nd edn., McGraw-Hill, London.

Avison, D., Lau, F., Myers, M., D. and Nielsen, P. A. (1999), "Action Research", *Communications of the Association for Computing Machinery,* 42(1), pp. 94-97.

Avital, M. and Vandenbosch, B. (2000), "SAP Implementation at Metalica: An Organizational Drama in Two Acts", *Journal of Information Technology,* 15(4), pp. 183-194.

Bansler, J. and Havn, E. C. (1994), "Information Systems Development with Generic Systems", in Baets, W. R. J. (Ed.) *Proceedings of the 2nd European Conference on Information Systems* Nijenrode University Press: Breukelen, pp. 707-715.

Banville, C. and Landry, M. (1992), "Can the Field of MIS be Disciplined", in Galliers, R. (Ed.) *Information Systems Research: Issues, Methods and Practical Guidelines*, Blackwell Scientific Publications, Oxford, pp. 61-88.

Bashein, B. J., Markus, M. L. and Riley, P. (1994), "Preconditions for BPR Success - And How to Prevent Failures", *Information Systems Management,* 11(2), pp. 7-13.

Baskerville, R. (1999), "Investigating Information Systems with Action Research", *Communications of the Association for Information Systems [http://www.cais.isworld.org],* 2(19), pp. 1-32.

Baskerville, R. and Wood-Harper, A. T. (1996), "A Critical Perspective on Action Research as a Method for Information Systems Research", *Journal of Information Technology,* 11(3), pp. 235-246.

Bennett, K. (1994), "Legacy Systems: Coping with Success", *IEEE Software,* 12(1), pp. 19-23.

Bhattacherjee, A. (2000), "Beginning SAP R/3 Implementation at Geneva Pharmaceuticals", *Communications of the Association for Information Systems [http://www.cais.isworld.org],* 4(2), pp. 1-39.

Bicknell, D. (1998), "SAP to fight drug firm's $500M. suit over R/3 collapse", *Computer Weekly,* 3 September, p. 3.

Bingi, P., Sharma, M. K. and Godla, J. K. (1999), "Critical Issues Affecting an ERP Implementation", *Information Systems Management,* 16(3), pp. 7-14.

Bloomfield, B., P. (1992), "Understanding the Social Practices of Systems Developers", *Journal of Information Systems,* 2, pp. 189-206.

Blum, B. (1996), *Beyond Programming: To a New Era of Design*, Oxford.

Bocij, P., Chaffey, D., Greasley, A. and Hickie, S. (2003), *Business Information Systems: Technology Development and Management for the E-Business,* Financial Times/Prentice Hall, Harlow.

Braa, K. and Vidgen, R. (1999), "Interpretation, Intervention, and Reduction in the Organizational Laboratory: A Framework for In-context Information System Research", *Accounting, Management and Information Technologies,* 9(1), pp. 25-47.

Brancheau, J. C., Janz, B. D. and Wetherbe, J. C. (1996), "Key Issues in Information Systems Management: 1994-95 SIM Delphi Results", *Management Information Systems Quarterly,* 20(2), pp. 225-242.

Brehm, L., Heinzl, A. and Markus, M. L. (2001), "Tailoring ERP Systems: A Spectrum of Choices and their Implications", in *Proceedings of the 34th Hawaii International Conference on System Sciences* IEEE Press: Maui, Hawaii, CD-ROM.

Brooke, C. (2002), "What Does it Mean to be 'Critical' in IS Research?" *Journal of Information Technology,* 17(2), pp. 49-57.

Brouthers, K. D. and van't Kruis, Y. M. (1997), "Competing in Software: Strategies for Europe's Niche Businesses", *Long Range Planning,* 30(4), pp. 518-528.

Brown, C. V. and Vessey, I. (2001), "NIBCO'S "Big Bang"", *Communications of the Association for Information Systems [http://www.cais.isworld.org],* 5(1), pp. 1-42.

Brown, E. (1998), "V F Corp. Changes Its Underware", *Fortune,* 7 December, pp. 79-82.

Brunsson, N. (1994), "The Virtue of Irrationality - Decision Making, Action and Commitment", in Clark, H., Chandler, J. and Barry, J. (Eds), *Organisation and Identities: Text and Readings in Organisational Behaviour*, Chapman and Hall, London, pp. 294-300.

Brunsson, N. (2000), *The Irrational Organization: Irrationality as a Basis for Organizational Change and Action,* Fagbokforlaget, Bergen.

Bryant, A. (1998), "Beyond BPR - Confronting the Organizational Legacy", *Management Decision,* 36(1), pp. 25-30.

Buchanan, D., Boddy, D. and McCalman, J. (1988), "Getting In, Getting On, Getting Out and Getting Back", in Bryam, A. (Ed.) *Doing Research in Organizations*, Routledge, London, pp. 53-67.

Bullock, A., Stallybrass, O. and Trombley, S. (Eds) (1988), *The Fontana Dictionary of Modern Thought,* 2nd edn.*,* Fontana Press, London.

Burrell, G. and Morgan, G. (1979), *Sociological Paradigms and Organisational Analysis,* Heinemann, London.

Business Week (1984), "Software: The New Driving Force", *Business Week,* 27 February, p. 74.

Butler, J. (1999), "Risk Management Skills Needed in a Packaged Software Environment", *Information Systems Management,* 16(3), pp. 15-20.

Buttle, F. (1995), *Relationship Marketing: An Overview (Working Paper No. 305),* Manchester Business School, Manchester.

Campbell, J. P., Daft, R. L. and Hulin, C. L. (1982), "Antecedents and Characteristics of Significant and Not-So-Significant Organizational Research", in Campbell, J. P. e. a. (Ed.) *What to Study: Generating and Developing Research Questions*, Sage, London, pp. 91-114.

Carmel, E. (1993), "How Quality Fits Into Packaged Development", *IEEE Software,* 10(5), pp. 85-86.

Carmel, E. (1995a), "Cycle Time in Packaged Software Firms", *Journal of Product Innovation Management,* 12(2), pp. 110-123.

Carmel, E. (1995b), "Time-to-Completion Factors in Packaged Software Development", *Information and Software Technology,* 37(9), pp. 515-520.

Carmel, E. (1997), "American Hegemony in Packaged Software Trade and the "Culture of Software"", *The Information Society,* 13(1), pp. 125-142.

Carmel, E. and Becker, S. (1995), "A Process Model for Packaged Software Development", *IEEE Transactions on Engineering Management,* 42(1), pp. 50-61.

Carmel, E. and Sawyer, S. (1998), "Packaged Software Development Teams: What Makes Them Different?" *Information Technology and People,* 11(1), pp. 7-19.

Champy, J. (1997), "Packaged Systems: One Way to Force Change", *Computerworld,* http://www.computerworld.com, Accessed: 19 August 2002.

Chau, P. Y. K. (1994), "Selection of Packaged Software in Small Businesses", *European Journal of Information Systems,* 3(4), pp. 292-302.

Chau, P. Y. K. (1995), "Factors Used in the Selection of Packaged Software in Small Businesses: Views of Owners and Managers", *Information and Management,* 29(2), pp. 71-78.

Checkland, P. (1981), *Systems Thinking, Systems Practice,* John Wiley and Sons, London.

Ciborra, C. and Associates (2000), *From Control to Drift: The Dynamics of Corporate Information Infrastructures,* Oxford University Press, Oxford.

Cliffe, S. (1999), "ERP Implementation", *Harvard Business Review,* 77(1), pp. 16-17.

Cohen, M. D., March, J. G. and Olsen, J. P. (1972), "A Garbage Can Model of Decision Making", *Administrative Science Quarterly,* 17(1), pp. 1-25.

Currie, W. (1995), *Management Strategy for IT: An International Perspective,* Pitman Publishing, London.

Currie, W. and Willcocks, L. (1998), "Analysing Four Types of IT Sourcing Decisions in the Context of Scale, Client/Supplier Interdependency and Risk Mitigation", *Information Systems Journal,* 8(2), pp. 119-143.

Curtis, B., Krasner, H. and Iscoe, N. (1988), "A Field Study of the Software Design Process for Large Systems", *Communications of the Association for Computing Machinery,* 31(11), pp. 1268-1287.

Damsgaard, J. and Truex III, D. P. (2000), "Binary Trading Relations and the Limits of EDI Standards: the Procrustean Bed of Standards", *European Journal of Information Systems,* 9(3), pp. 173-188.

Dautermann, J. (1990), "Putting a Local Information System Online Using Pre-packaged Software", in Talburt, J. R. (Ed.) *Proceedings of the Annual ACM Conference on Systems Documentation* ACM Press: Little Rock, USA, pp. 49-53.

Davenport, T. H. and Markus, M. L. (1999), "Rigor vs. Relevance Revisited: Response to Benbasat and Zmud", *Management Information Systems Quarterly,* 23(1), pp. 19-23.

Davenport, T. H. and Short, J. E. (1990), "The New Industrial Engineering: Information Technology and Business Process Redesign", *Sloan Management Review,* 32(4), pp. 11-27.

Davenport, T. H. and Stoddard, D., B. (1994), "Reengineering; Business Change of Mythic Proportions", *Management Information Systems Quarterly,* 18(2), pp. 121-127.

Davis, R., Samuelson, P., Kapor, M. and Reichman, J. (1996), "A New View of Intellectual Property and Software", *Communications of the Association for Computing Machinery,* 39(3), pp. 21-30.

Deloitte and Touche (1996), "1996 CIO Survey: Major Packages", *Deloitte and Touche,* http://www.dttus.com, Accessed: 17 April 1998.

Denzin, N. K. (2001), *Interpretive Interactionism,* Sage Publications, Thousand Oaks.

Dolmetsch, R., Huber, T., Fleisch, E. and Osterle, H. (1998), *Accelerated SAP: 4 Case Studies,* IWI-HSG - Universitat St Gallen, St. Gallen.

Donoghue, R. G. (1983), "Installing Manufacturing System Software Package", *Journal of Systems Management,* 34(2), pp. 21-23.

Dube, L. (1998), "Teams in Packaged Software Development: The Software Corp. Experience", *Information Technology and People,* 11(1), pp. 36-61.

Economist (1994), "Europe's Software Debacle", *Economist,* 12 November, pp. 101-102.

Edwards, J., Coutts, I., McLeod, S. and Millea, T. (2000), "Handling Legacy IT in Banking by Using Object Design Patterns to Separate Business and IT Issues", in Henderson, P. (Ed.) *Systems Engineering for Business Process Change: Collected Papers from the EPSRC Research Programme*, Springer, London, pp. 222-238.

Eisenhardt, K. M. (1989), "Building Theories from Case Study Research", *Academy of Management Review,* 14(4), pp. 532-550.

Eisenhardt, K. M. (1991), "Better Stories and Better Constructs: The Case for Rigor and Comparative Logic", *Academy of Management Review,* 16(3), pp. 620-627.

Esteves, J. and Pastor, J. (2001), "Enterprise Resource Planning Systems Research: An Annotated Bibliography", *Communications of the Association for Information Systems [http://www.cais.isworld.org],* 7(8), pp. 1-52.

Esteves, J. M. and Pastor, J. A. (1999), "An ERP Life-Cycle-Based Research Agenda", in Eder, J., Maiden, N. and Missikoff, M. (Eds), *Proceedings of the First International Workshop in Enterprise Management and Resource Planning: Methods, Tools and Architectures* Centro de Studi San Salvador: Venice, CD-ROM.

Fan, M., Stallaert, J. and Whinston, A. B. (2000), "The Adoption and Design Methodologies of Component-Based Enterprise Systems", *European Journal of Information Systems,* 9(1), pp. 25-35.

Fay, B. (1987), "An Alternative View: Interpretive Social Science", in Gibbons, M. (Ed.) *Interpreting Politics*, Basil Blackwell Ltd, Oxford, pp. 82-100.

Ferneley, E. and Light, B. (2002), "Knowledge Management Systems: The Role of 'Encouragement'", in Wrycza, S. (Ed.) *Proceedings of the Xth European Conference on Information Systems* University of Gdansk: Gdansk, Poland, pp. 884-893.

Fitzgerald, B. (1996), "Formalised Systems Development Methodologies: A Critical Perspective," *Information Systems Journal,* 6(1), pp. 3-23.

Fitzgerald, B., Russo, N. L. and Stolterman, E. (2002), *Information Systems Development: Method in Action,* McGraw Hill Companies, London.

Flowers, S. (1996), *Software Failure: Management Failure,* John Wiley and Sons, Chichester.

Flynn, D. (1998), *Information Systems Requirements: Determination and Analysis,* The McGraw-Hill Companies, London.

Flynn, D. and Davarpanah Jazi, M. (1998), "Constructing User Requirements: A Social Process for a Social Context", *Information Systems Journal,* 8(1), pp. 53-83.

Forrester Research (1998), "Sizing Commerce Software", *Forrester Research,* http://www.forrester.com, Accessed: 6 June 1999.

Friedman, A. L. and Cornford, D. S. (1989), *Computer Systems Development: History, Organization and Implementation,* John Wiley and Sons, Chichester.

Gable, G., G. (1994), "Integrating Case Study and Survey Research Methods: An Example in Information Systems", *European Journal of Information Systems,* 3(2), pp. 112-126.

Galliers, R. D. and Land, F. (1987), "Choosing Appropriate Information System Research Methodologies", *Communications of the Association for Computing Machinery,* 30(11), pp. 900-902.

Galliers, R.D. and Newell, S. (2001), "Back to the Future: From Knowledge Management to Data Management", in Smithson, S., Cricar, J., Podlogar, M. and Avgerinou, S. (Eds), *Proceedings of the 9th European Conference on Information Systems* University of Maribor: Bled, Slovenia, pp. 609-615.

Gefen, D. (2002), "Nurturing Clients' Trust to Encourage Engagement Success During the Customization of ERP Systems", *Omega - The International Journal of Management Science,* 30(4), pp. 287-299.

Ghauri, P. N., Gronhaug, K. and Kristianslund, I. (1995), *Research Methods In Business Studies: A Practical Guide,* Prentice Hall, New York.

Giaglis, G. M., Klein, S. and O'Keefe, R. (2002), "The Role of Intermediaries in Electronic Marketplaces: Developing a Contingency Model", *Information Systems Journal,* 12(3), pp. 231-246.

Gibbons, M. (1987), "Introduction: The Politics of Interpretation", in Gibbons, M. (Ed.) *Interpreting Politics*, Basil Blackwell Ltd, Oxford, pp. 1-29.

Gibson, C. F. and Nolan, R. L. (1974), "Managing the Four Stages of EDP Growth", *Harvard Business Review,* 52(1), pp. 76-88.

Gibson, N., Holland, C. and Light, B. (1999), "A Fast Track SAP R/3 Implementation at Guilbert Niceday", *Electronic Markets,* 9(3), pp. 190-193.

Gill, J. and Johnson, P. (1991), *Research Methods for Managers,* Paul Chapman Publishing, London.

Glass, R. L. (1998), "Enterprise Resource Planning - Breakthrough and/or Term Problem?" *Data Base for Advances in Information Systems,* 29(2), pp. 14-16.

Goles, T. and Hirschheim, R. (2000), "The paradigm is dead, the paradigm is dead...long live the paradigm; the legacy of Burrell and Morgan." *Omega - The International Journal of Management Science,* 28(3), pp. 249-268.

Golland, M. L. (1978), "Buying or Making the Software Package That is Best for You", *Journal of Systems Management,* 29(8), pp. 48-51.

Goodley, S. and Bennett, E. (2000), "Hey Big Vendor", *Computing2,* June, pp. 42-46.

Graziano, A. M. and Raulin, M. L. (1997), *Research Methods: A Process of Enquiry,* 3rd edn., Longman, New York.

Gremillion, L. L. (1982), "Improving Productivity with Application Software Packages", *Business Horizons,* 25(2), pp. 51-54.

Gronroos, C. (1994), "Quo Vadis, Marketing? Toward a Relationship Marketing Paradigm", *Journal of Marketing Management,* 10(5), pp. 347-360.

Gross, P. H. B. and Ginzberg, M. J. (1984), "Barriers to the Adoption of Application Software Packages", *Systems, Objectives, Solutions,* 4(4), pp. 211-226.

Guba, E. G. and Lincoln, Y. S. (1994), "Competing Paradigms in Qualitative Research", in Denzin, N. K. and Lincoln, Y. S. (Eds), *Handbook of Qualitative Research*, Sage Publications Inc, Thousand Oaks, pp. 105-117.

Gummesson, E. (1991), *Qualitative Methods in Management Research,* Sage Publications Inc, Newbury Park.

Hall, G., Rosenthal, J. and Wade, J. (1993), "How to Make Reengineering Really Work", *Harvard Business Review,* 71(6), pp. 119-131.

Hammer, M. (1990), "Reengineering Work: Don't Automate, Obliterate", *Harvard Business Review,* 68(4), pp. 104-112.

Hammer, M. and Champy, J. (1994), *Reengineering the Corporation,* Harper Business, New York.

Hanseth, O. and Braa, K. (1998), "Technology as Traitor: Emergent SAP Infrastructure in a Global Organization", in Hirschheim, R., Newman, M. and De Gross, J. I. (Eds), *Proceedings of the 19th International Conference on Information Systems* Association for Information Systems: Helsinki, Finland, pp. 188-196.

Hanseth, O., Ciborra, C. and Braa, K. (2001), "The Control Devolution: ERP and Side Effects of Globalization", *The Data Base for Advances in Information Systems,* 32(4), pp. 34-46.

Hart, C. (1998), *Doing a Literature Review,* Sage, London.

Hecht, B. (1997), "Choosing the Right ERP Software", *Datamation,* March, pp. 56-58.

Hedrick, T. E., Bickman, L. and Rog, D. J. (1993), *Applied Research Design: A Practical Guide,* Sage Publications Ltd, London.

Heikkila, J., Saarinen, T. and Saaksjarvi, M. (1991), "Success of Software Packages in Small Businesses: An Exploratory Study", *European Journal of Information Systems,* 1(3), pp. 159-169.

Henderson, P. (2000a), "Business Processes, Legacy Systems and a Flexible Future", in Henderson, P. (Ed.) *Systems Engineering for Business Process Change: Collected Papers from the EPSRC Research Programme*, Springer, London, pp. 1-9.

Henderson, P. (Ed.) (2000b), *Systems Engineering for Business Process Change: Collected Papers from the EPSRC Research Programme,* Springer, London.

Hirt, S. G. and Swanson, E. B. (1999), "Adopting SAP at Siemens Power Corporation", *Journal of Information Technology,* 14(3), pp. 243-251.

Holland, C. and Light, B. (1999a), "Global Enterprise Resource Planning Implementation", in *Proceedings of the 32nd Annual Hawaii International Conference on System Sciences* IEEE Computer Society Press: Hawaii, CD-ROM.

Holland, C. and Light, B. (2001), "A Stage Maturity Model for Enterprise Resource Planning Systems", *Data Base for Advances in Information Systems,* 32(2), pp. 34-45.

Holland, C., Light, B. and Gibson, N. (1998), "Global Enterprise Resource Planning Implementation", in Hoadley, E. D. and Benbasat, I. (Eds), *Proceedings of the 4th Americas Conference on Information Systems* Association For Information systems: Baltimore, USA, pp. 421-423.

Holland, C., Light, B. and Gibson, N. (1999a), "A Critical Success Factors Model for Enterprise Resource Planning Implementation", in *Proceedings of the 7th European Conference on Information Systems*, Vol. 1 Copenhagen, pp. 273-287.

Holland, C., Light, B. and Kawalek, P. (1999b), "Beyond Enterprise Resource Planning Projects: Innovative Strategies for Competitive Advantage", in Pries-Heje, J., Ciborra, C., Kautz, K., Valor, J., Christiaanse, E., Avison, D. and Heje, C. (Eds), *Proceedings of the 7th European Conference on Information Systems*, Vol. 1 Copenhagen Business School: Copenhagen, Denmark, pp. 288-301.

Holland, C., Lockett, G., Richard, J. M. and Blackman, I. (1994), "The Evolution of a Global Cash Management System", *Sloan Management Review,* 36(1), pp. 37-47.

Holland, C. P., Blackman, I., Light, B. and Gibson, N. (1999c), *An Analysis of Business Strategies to Overcome the Year 2000 Problem in International Banking,* Warwick Working Paper Series, Warwick University, Warwick.

Holland, C. P. and Light, B. (1999b), "A Critical Success Factors Model for ERP Implementation", *IEEE Software,* 16(3), pp. 30-36.

Holland, C. P. and Light, B. (2003), "A Framework for Understanding Success and Failure in ERP Implementation", in Shanks, G., Seddon, P. and Willcocks, L. (Eds), *Enterprise Systems: ERP, Implementation and Effectiveness (Forthcoming)*, Cambridge University Press, Cambridge.

Holmström, H. (2001), "Virtual Communities as Platforms for Product Development: An Interpretive Case Study of Customer Involvement in Online Game Development", in Sarkar, S., Storey, V. C. and De Gross, J. I. (Eds), *Proceedings of the 22nd International Conference on Information Systems* Association for Information Systems: New Orleans, USA, pp. 299-306.

Howcroft, D. (2001), "After the Goldrush: Deconstructing the Myths of the Dot.com Market", *Journal of Information Technology,* 16(4), pp. 195-204.

Howcroft, D. and Light, B. (2002), "A Study of User Involvement in Packaged Software Selection", in Applegate, L., Galliers, R. D. and De Gross, J. I. (Eds), *Proceedings of the 23rd International Conference on Information Systems* Association for Information Systems: Barcelona, Spain, pp. 69-77.

Huber, T., Alt, R. and Osterle, H. (2000), "Templates - Instruments for Standardizing ERP Systems", in *Proceedings of the 33rd Hawaii International Conference on System Sciences* IEEE Press: Maui, Hawaii, CD-ROM.

Janson, M. A. and Subramanian, A. (1995), "Packaged Software: Selection and Implementation Policies", *Infor,* 34(2), pp. 133-151.

Jennings, D., Lye, R., Pizzey, A., Stewart, J., Wattam, S. and Whysall, P. (1994), "Developing Skills in Decision Making", in Jennings, D. and Wattam, S. (Eds), *Decision Making: An Integrated Approach*, Pitman, London, pp. 283-309.

Jennings, D. and Wattam, S. (1994), "Decision Making", in Jennings, D. and Wattam, S. (Eds), *Decision Making: An Integrated Approach*, Pitman, London, pp. 1-26.

Kaasbøll, J. J. (1997), "How Evolution of Information Systems May Fail: Many Improvements Adding Up to Negative Effects", *European Journal of Information Systems,* 6(3), pp. 172-180.

Kahneman, D. and Tversky, A. (1983), "Choices, Values and Frames", *American Psychologist,* 39(4), pp. 341-350.

Kaplan, B. and Duchon, D. (1988), "Combining Qualitative and Quantitative Methods in Information Systems Research", *Management Information Systems Quarterly,* 12(4), pp. 571-586.

Kawalek, P. and Leonard, J. (1996), "Evolutionary Software Development to Support Organizational and Business Process Change: A Case Study Account", *Journal of Information Technology,* 11(3), pp. 185-198.

Kearney Management Consultants (1984), *The Barriers and the Opportunities of Information Technology - a Management Perspective,* The Institute of Administrative Management, Orpington.

Keil, M. and Carmel, E. (1995), "Customer-Developer Links in Software Development", *Communications of the Association for Computing Machinery,* 38(5), pp. 33-44.

Kelly, S., Holland, C., Gibson, N. and Light, B. (1999a), "A Business Perspective of Legacy Systems", *Communications of the Association for Information Systems [http://www.cais.isworld.org],* 2(9), pp. 1-27.

Kelly, S., Holland, C. and Light, B. (1999b), "A Departure from Traditional Systems Development Methodologies: Enterprise Resource Planning (ERP) Systems and the Use of Process Modelling Tools Manchester", in Hackney, R. and Dunn, D. (Eds), *Proceedings of the 9th Annual BIT Conference* Manchester Metropolitan University: Manchester, UK, CD-ROM.

Kim, Y.-G. (1997), "Improving Legacy Systems Maintainability", *Information Systems Management,* 14(1), pp. 7-11.

Klaus, H., Rosemann, M. and Gable, G. G. (2000), "What is ERP?" *Information Systems Frontiers,* 2(2), pp. 141-162.

Klein, H., K. and Myers, M., D. (1999), "A Set of Principles for Conducting and Evaluating Interpretive Field Studies in Information Systems", *Management Information Systems Quarterly,* 23(1), pp. 67-94.

Klepper, R. and Hartog, C. (1992), "Trends in the Use and Management of Application Package Software", *Information Resources Management Journal,* 5(4), pp. 33-37.

Kling, R. and Iacono, S. (1984), "The Control of Information Systems Developments After Implementation", *Communications of the Association for Computing Machinery,* 27(12), pp. 1218-1226.

Knights, D., Noble, F. and Willmott, H. (1997), "'We Should be Total Slaves to the Business': Aligning Information Technology and Strategy - Issues and Evidence", in Bloomfield, B. P., Coombs, R., Knights, D. and Littler, D. (Eds), *Information Technology and Organizations: Strategies, Networks and Integration*, Oxford University Press, Oxford, pp. 13-35.

Knowles, J. (1997), "Buy Versus Build in Six Words", *Datamation,* January, p. 31.

Kotamraju, N. (2002), "Keeping Up: Web Design Skill and the Reinvented Worker", *Information Communication and Society,* 5(1), pp. 1-26.

KPMG (1998), *Exploiting Packaged Software,* KPMG, London.

Kraemmergaard, P. and Rose, J. (2002), "Managerial Competences for ERP Journeys", *Information Systems Frontiers,* 4(2), pp. 199-211.

Krishnan, M. S. (1998), "The Role of Team Factors in Software Cost and Quality", *Information Technology and People,* 11(1), pp. 20-35.

Krumbholz, M., Galliers, J. and Coulianos, N. (2000), "Implementing Enterprise Resource Planning Packages in Different Corporate and National Cultures", *Journal of Information Technology,* 15(4), pp. 267-279.

Kunda, D. and Brooks, L. (2000), "Identifying and Classifying Processes (Traditional and Soft Factors) that Support COTS Component Selection: A Case Study", *European Journal of Information Systems,* 9(4), pp. 226-234.

Lai, L. S. L. (1998), "An Expectation-Perception Gap Analysis of Information Systems Failure", in Wood-Harper, A. T., Jayaratna, N. and Wood, J. R. G. (Eds), *Methodologies for Developing and Managing Emerging Technology Based Systems, Proceedings of the 6th International Conference of the British Computer Society Information Systems Methodologies Specialist Group* Springer: Salford, pp. 130-141.

Lassila, K. S. and Brancheau, J. C. (1999), "Adoption and Utilization of Commercial Software Packages: Exploring Utilization Equilibria, Transitions, Triggers and Tracks", *Management Information Systems Quarterly,* 16(2), pp. 63-90.

Lau, F. (1999), "Toward a Framework for Action Research in Information Systems Studies", *Information Technology and People,* 12(2), pp. 148-175.

Lauder, A. and Kent, S. (2000), "Legacy System Anti-Patterns and a Pattern-Oriented Migration Response", in Henderson, P. (Ed.) *Systems Engineering for Business Process Change: Collected Papers from the EPSRC Research Programme*, Springer, London, pp. 239-250.

Laudon, K. C. and Laudon, J. P. (2000), *Management Information Systems: Organization and Technology in the Networked Enterprise,* 6th edn., Prentice Hall Inc., New Jersey.

Lawler, E. E. (1985), "Challenging Traditional Research Assumptions", in Lawler, E. E., Mohrman, A. M., Mohrman, S. A., Ledford, G. E., Cummings, T. G. and Associates (Eds), *Doing Research that is Useful for Theory and Practice*, Jossey-Bass Ltd, Oxford, pp. 1-17.

Lee, A. S. (1991), "Integrating Positivist and Interpretive Approaches to Organizational Research", *Organization Science,* 2(4), pp. 342-365.

Lee, A. S. (1999a), "Researching MIS", in Currie, W. and Galliers, R. D. (Eds), *Rethinking Management Information Systems*, Oxford University Press, Oxford, pp. 7-27.

Lee, A. S. (1999b), "Rigor and Relevance in MIS Research: Beyond the Approach of Positivism Alone", *Management Information Systems Quarterly,* 23(1), pp. 29-33.

Lee, Z. and Lee, J. (2000), "An ERP Implementation Case Study from a Knowledge Transfer Perspective", *Journal of Information Technology,* 15(4), pp. 281-288.

Leebaert, D. (1995), "News from the Frontiers", in Leebaert, D. (Ed.) *The Future of Software*, MIT Press, Cambridge, pp. 1-28.

Lehman, M. M. (1980), "Life-Cycles, and the Laws of Software Evolution", *Proceedings of the IEEE,* 68(9), pp. 1060-1076.

Levina, N. (2002), "Collaborative Practices in Information Systems Development: A Collective Reflection-in-Action Framework", in Applegate, L., Galliers, R. D. and De Gross, J. I. (Eds), *Proceedings of the 23rd International Conference on Information Systems* Association for Information Systems: Barcelona, Spain, pp. 267-277.

Lévi-Strauss, C. (1966), *The Savage Mind (La Pensée Savage),*Weidenfeld and Nicolson, London.

Lewis, P. J. (1994), *Information Systems Development: Systems Thinking in the Field of IS,*Pitman, London.

Li, C. (1999), "ERP Packages: What's Next?" *Information Systems Management,* 16(3), pp. 31-35.

Light, B. (1999), "Realizing the Potential of ERP Systems", *Electronic Markets,* 9(4), pp. 238-241.

Light, B. (2000), "The Evolution of Business Process Reengineering", in Hackney, R. and Dunn, D. (Eds), *Business Information Technology: Alternative and Adaptive Futures*, MacMillan Press Ltd, London, pp. 291-306.

Light, B. (2001a), "The Maintenance Implications of the Customisation of ERP Software", *The Journal of Software Maintenance: Research and Practice,* 13(6), pp. 415-430.

Light, B. (2001b), "A Multi-Perspective Analysis of the Strategic Impact of Customer Relationship Management Systems", in Khosropour, M. (Ed.) *Proceedings of the Information Resources Management Association: International Conference* Idea Group Publishing: Toronto, Canada, pp. 1192-1193.

Light, B. (2001c), "A Review of the Issues Associated with Customer Relationship Management Systems", in Smithson, S., Cricar, J., Podlogar, M. and Avgerinou, S. (Eds), *Proceedings of the 9th European Conference on Information Systems* University of Maribor: Bled, Slovenia, pp. 1232-1241.

Light, B. (2003a), "An Alternative Theory of Legacy Information Systems", (Working Paper).

Light, B. (2003b), "A Study of Organisational Experiences of CRM Packaged Software (Forthcoming)", *Business Process Management Journal*.

Light, B. and Holland, C. (2000), "Enterprise Resource Planning Systems: Impacts and Future Directions", in Henderson, P. (Ed.) *Systems Engineering for Business Process Change: Collected Papers from the EPSRC Research Programme*, Springer, London, pp. 117-126.

Light, B., Holland, C., Kelly, S. and Wills, K. (2000), "Best of Breed IT Strategy: An Alternative to Enterprise Resource Planning Systems", in Robert Hansen, H., Bichler, M. and Mahrer, H. (Eds), *Proceedings of the 8th European Conference on Information Systems*, Vol. 1 Vienna University of Economics and Business Administration: Vienna, Austria, pp. 652-659.

Light, B., Holland, C. and Wills, K. (2001), "ERP and Best of Breed: A Comparative Analysis", *Business Process Management Journal,* 7(3), pp. 216-224.

Lloyd, C. and Newell, H. (1998), "Computerising the Sales Force: The Introduction of Technical Change in a Non-Union Workforce", *New Technology, Work and Employment,* 13(2), pp. 104-115.

Lockett, A. G., Barrar, P. R. N. and Polding, M. E. (1991), "MRP II Systems: Success Factors in the Process of Implementation", in Pridham, M. and O'Brien, C. (Eds), *Production Research: Approaching the 21st Century*, Taylor Francis, London, pp. 55-65.

Lucas, H. C. J., Walton, E. J. and Ginzberg, M. J. (1988), "Implementing Packaged Software", *Management Information Systems Quarterly,* 12(4), pp. 536-549.

Lye, R. (1994), "Group and Organisational Decision Making", in Jennings, D. and Wattam, S. (Eds), *Decision Making: An Integrated Approach*, Pitman, London, pp. 54-95.

Lynch, R. K. (1984), "Implementing Packaged Application Software: Hidden Costs and New Challenges", *Systems, Objectives, Solutions,* 4(4), pp. 227-234.

Lynch, R. K. (1987), "The Impact of Packaged Software on User/Vendor Life Cycle Concepts", *Journal of Information Systems Management,* 4(2), pp. 34-40.

MacNealy, M. S. (1997), "Toward Better Case Study Research", *IEEE Transactions on Professional Communication,* 40(3), pp. 182-196.

Markus, M. L. (1983), "Power, Politics, and MIS Implementation", *Communications of the Association for Computing Machinery,* 26(6), pp. 430-444.

Markus, M. L. (2000), "Conceptual Challenges in Contemporary IS Research", *Communications of the Association for Information Systems [http://www.cais.isworld.org],* 3(4).

Markus, M. L., Axline, S., Petrie, D. and Tanis, C. (2000), "Learning From Adopters' Experiences with ERP: Problems Encountered and Success Achieved", *Journal of Information Technology,* 15(4), pp. 245-265.

Markus, M. L. and Benjamin, R. I. (1997), "The Magic Bullet Theory of IT-enabled Transformation", *Sloan Management Review,* 38(2), pp. 55-68.

Markus, M. L. and Bjørn-Andersen, N. (1987), "Power Over Users: Its Exercise By System Professionals", *Communications of the Association for Computing Machinery,* 30(6), pp. 498-504.

Markus, M. L. and Lee, A. S. (1999), "Special Issue on Intensive Research in Information Systems: Using Qualitative, Interpretive, and Case Methods to Study Information Technology - Foreword", *Management Information Systems Quarterly,* 23(1), pp. 35-38.

Markus, M. L. and Tanis, C. (2000), "The Enterprise System Experience - From Adoption to Success", in Zmud, R. W. (Ed.) *Framing the Domains of IT Research: Glimpsing the Future Through the Past*, Pinnaflex Educational Resources, Cincinnati, pp. 173-207.

Martin, J. and McClure, C. (1983), "Buying Software Off the Rack", *Harvard Business Review,* 61(6), pp. 32-62.

Martin, M. H. (1998), "An ERP Strategy", *Fortune,* 2 February, pp. 95-97.

Martinez, E., V. (1995), "Successful Reengineering Demands IS/Business Partnerships", *Sloan Management Review,* 37(4), pp. 51-60.

McClintock, C. C., Brannon, D. and Maynard-Moody, S. (1979), "Applying the Logic of Sample Surveys to Qualitative Case Studies: The Case Cluster Method", *Administrative Science Quarterly,* 24(4), pp. 612-629.

McKay, J. and Marshall, P. (2001), "The Dual Imperatives of Action Research", *Information Technology and People,* 14(1), pp. 46-59.

McKeen, J., Smith, H., Joglekar, N. and Balasubramanian, P. R. (2002), "Developments in Practice V: IT Sourcing: Build, Buy or Market", *Communications of the Association for Information Systems [http://www.cais.isworld.org],* 9(8), pp. 120-135.

Miles, M. B. (1979), "Qualitative Data as an Attractive Nuisance: The Problem of Analysis", *Administrative Science Quarterly,* 24(4), pp. 590-601.

Miles, M. B. and Huberman, A. M. (1994), *Qualitative Data Analysis: An Expanded Sourcebook,* 2nd edn., Sage Publications Inc, Thousand Oaks.

Miller, D. C. (1991), *Handbook Of Research Design And Social Measurement,* 5th edn., Sage, London.

Mingers, J. (2001), "Combining IS Research Methods: Towards a Pluralist Methodology", *Information Systems Research,* 12(3), pp. 240-259.

Mitev, N. (1999), "Electronic Markets in Transport: Comparing the Globalization of Air and Rail Computerized Reservation Systems", *Electronic Markets,* 9(4), pp. 215-225.

Mitroff, I. I. (1985), "Why Our Old Pictures of the World Do Not Work Anymore", in Lawler, E. E., Mohrman, A. M., Mohrman, S. A., Ledford, G. E., Cummings, T. G. and Associates (Eds), *Doing Research that is Useful for Theory and Practice*, Jossey-Bass Ltd, Oxford, pp. 18-35.

Montazemi, A. R., Cameron, D. A. and Gupta, K. M. (1996), "An Empirical Study of Factors Affecting Software Package Selection", *Journal of Management Information Systems,* 13(1), pp. 89-105.

Mumford, E. (2001), "Advice for an Action Researcher", *Information Technology and People,* 14(1), pp. 12-27.

Myers, M., D. (1997), "Interpretive Research in Information Systems", in Mingers, J. and Stowell, F. (Eds), *Information Systems: An Emerging Discipline*, The McGraw Hill Companies, London, pp. 239-266.

Myers, M., D. (1999), "Investigating Information Systems with Ethnographic Research", *Communications of the Association for Information Systems [http://www.cais.isworld.org],* 2(23), pp. 1-20.

Nandhakumar, J. and Jones, M. (1997), "Too Close for Comfort? Distance and Engagement in Interpretive Information Systems Research", *Information Systems Journal,* 7(2), pp. 109-131.

Nassif, R. and Mitchusson, D. (1993), "Issues and Approaches for Migration/Cohabitation Between Legacy and New Systems", in *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data* ACM Press, NY, pp. 471-474.

Nelson, C. N., Treichler, P. A. and Grossberg, L. (1992), "Cultural Studies: An Introduction", in Grossberg, L., Nelson, C. N. and Treichler, P. A. (Eds), *Cultural Studies*, Routledge, New York, pp. 1-22.

Nelson, P., Richmond, W. and Seidmann, A. (1996), "Two Dimensions of Software Acquisition", *Communications of the Association for Computing Machinery,* 39(7), pp. 29-35.

Newell, S., Huang, J. C. and Tansley, C. (2002), "Social Capital in ERP Projects: The Differential Source and Effects of Bridging and Bonding", in Applegate, L., Galliers, R. D. and De Gross, J. I. (Eds), *Proceedings of the 23rd International Conference on Information Systems* Association for Information Systems: Barcelona, Spain, pp. 257-265.

Newell, S., Robertson, M. and Swan, J. (2001), "Management Fads and Fashions", *Organization,* 8(1), pp. 5-15.

Nolan, R. L. (1973), "Managing the Computer Resource: A Stage Hypothesis", *Communications of the Association for Computing Machinery,* 16(7), pp. 399-405.

Numagami, T. (1998), "The Infeasibility of Invariant Laws in Management Studies: A Reflective Dialogue in Defense of Case Studies", *Organization Science,* 9(1), pp. 2-15.

OECD (2000), *OECD Information Technology Outlook,* OECD, Paris.

OECD (2002), *OECD Information Technology Outlook (Highlights),* OECD, Paris.

Oliver, D. and Romm, C. (2000), "ERP Systems: The Route to Adoption", in Chung, H. M. (Ed.) *Proceedings of the 6th Americas Conference on Information Systems* Association for Information Systems: Long Beach, USA, pp. 1039-1044.

Orenstein, D. (1998), "Retailers Seek More ERP Functionality", http://www.computerworld.com, Accessed: 2 November 1998.

Orlikowski, W. J. and Baroudi, J. J. (1991), "Studying Information Technology in Organizations: Research Approaches and Assumptions", *Information Systems Research*, 2(1), pp. 1-28.

Pan, S. L., Huang, J. C., Newell, S. and Cheung, A. (2001), "Knowledge Integration as a Key Problem in An ERP Implementation", in Sarkar, S., Storey, V. C. and De Gross, J. I. (Eds), *Proceedings of the 22nd International Conference on Information Systems* Association for Information Systems: New Orleans, USA, pp. 321-327.

Parr, A. and Shanks, G. (2000), "A Model of ERP Project Implementation", *Journal of Information Technology,* 15(4), pp. 289-303.

Paul, R. J. (2002), "$(IS)^3$: Is Information Systems and Intellectual Subject?" *European Journal of Information Systems,* 11(2), pp. 174-177.

Payne, A., Christopher, M., Clark, M. and Peck, H. (1999), *Relationship Marketing for Competitive Advantage,* 2nd edn., Butterworth Heinemann, Oxford.

Peppard, J. (2001), "Bridging the Gap Between the IS Organization and the Rest of the Business: Plotting a Route", *Information Systems Journal,* 11(3), pp. 249-270.

Pettigrew, A. M. (1985), "Contextualist Research: A Natural Way to Link Theory and Practice", in Lawler, E. E., Mohrman, A. M., Mohrman, S. A., Ledford, G. E., Cummings, T. G. and Associates (Eds), *Doing Research that is Useful for Theory and Practice*, Jossey-Bass Ltd, Oxford, pp. 222-274.

Pinto, J., K. and Slevin, D., P. (1987), "Critical Factors in Successful Project Implementation", *IEEE Transactions on Engineering Management,* 34(1), pp. 22-27.

Porter, M. (1980), *Competitive Strategy,* Free Press, New York.

Porter, M., E. (1996), "What is Strategy", *Harvard Business Review,* 74(6), pp. 61-78.

Pozzebon, M. (2001), "Demystifying the Rhetorical Closure of ERP Packages", in Sarkar, S., Storey, V. C. and De Gross, J. I. (Eds), *Proceedings of the 22nd International Conference on Information Systems* Association for Information Systems: New Orleans, USA, pp. 329-337.

Pressman, R. S. (1997), *Software Engineering: A Practitioner's Approach,* 4th edn., The McGraw-Hill Companies Inc., New York.

PriceWaterhouse (1996), *PriceWaterhouse Information Technology Review 1995/1996,* PriceWaterhouse, London.

Punch, K. F. (2000), *Developing Effective Research Proposals,* Sage Publications Ltd, London.

Quintas, P. (1994), "Programmed Innovation? Trajectories of Change in Software Development", *Information Technology and People,* 7(1), pp. 25-47.

Raghunathan, S. (2000), "Software Editions: An Application of Segmentation Theory to the Packaged Software Market", *Journal of Management Information Systems,* 17(1), pp. 87-113.

Ramage, M., Brooke, C., Bennett, K. and Munro, M. (2000), "Combining Organisational and Technical Change in Finding Solutions to Legacy Systems", in Henderson, P. (Ed.) *Systems Engineering for Business Process Change: Collected Papers from the EPSRC Research Programme*, Springer, London, pp. 79-90.

Rapoport, R. N. (1970), "Three Dilemmas in Action Research", *Human Relations,* 23(6), pp. 499-513.

Remenyi, D., Sherwood-Smith, M. and White, T. (1997), *Achieving Maximum Value From Information Systems: A Process Approach,* John Wiley and Sons, Chichester.

Remenyi, D. and Williams, B. (1996), "The Nature of Research: Qualitative or Quantitative, Narrative or Paradigmatic?" *Information Systems Journal,* 6(2), pp. 131-146.

Robey, D. and Markus, M. L. (1984), "Rituals in Information System Design", *Management Information Systems Quarterly,* 8(1), pp. 5-15.

Robey, D. and Boudreau, M.C. (1999), "Accounting for the Contradictory Organizational Consequences of Information Technology: Theoretical Directions and Methodological Implications", *Information Systems Research*, 10(2), pp. 167-185.

Robson, W. (1997), *Strategic Management and Information Systems,* 2nd edn., Pitman Publishing, London.

Rollinson, D. and Broadfield, A. (2002), *Organisational Behaviour and Analysis,* 2nd edn., Financial Times/Prentice Hall, Harlow.

Ross, J. W. (1999), "Dow Corning Corporation: Business Processes and Information Technology", *Journal of Information Technology,* 14(3), pp. 253-266.

Ross, J. W. and Vitale, M. R. (2000), "The ERP Revolution: Surviving vs. Thriving", *Information Systems Frontiers,* 2(2), pp. 233-241.

Sawyer, S. (2000), "Packaged Software: Implications of the Differences from Custom Approaches to Software Development", *European Journal of Information Systems,* 9(1), pp. 47-58.

Sawyer, S. (2001), "A Market-Based Perspective on Information Systems Development", *Communications of the Association for Computing Machinery,* 44(11), pp. 97-102.

Sayer, A. (1992), *Method In Social Science,* 2nd edn., Routledge, London.

Scheer, A. W. and Habermann, F. (2000), "Making ERP a Success", *Communications of the Association for Computing Machinery,* 43(4), pp. 57-61.

Schmidt, S. K. and Werle, R. (1998), *Coordinating Technology: Studies in the International Standardization of Telecommunications,* The MIT Press, Cambridge.

Schwandt, T. A. (1994), "Constructivist, Interpretivist Approaches to Human Inquiry", in Denzin, N. K. and Lincoln, Y. S. (Eds), *Handbook of Qualitative Research*, Sage Publications Inc, Thousand Oaks, pp. 118-137.

Schwandt, T. A. (2001), *Dictionary of Qualitative Inquiry,* Sage Publications, Thousand Oaks.

Scott, J., E. and Kaindl, L. (2000), "Enhancing Functionality in an Enterprise Software Package", *Information and Management,* 37(3), pp. 111-122.

Scott, J., E. and Vessey, I. (2000), "Implementing Enterprise Resource Planning Systems: The Role of Learning from Failure", *Information Systems Frontiers,* 2(2), pp. 213-232.

Sharland, R. (1991), *Package Evaluation: A Practical Guide to Selecting Application and Systems Software,* Avebury Technical, Aldershot.

Sherer, S. A. (1993), "Purchasing Software Systems: Managing the Risk", *Information and Management,* 24(5), pp. 257-266.

Sieber, T., Siau, K., Nah, F. and Sieber, M. (2000), "SAP Implementation at the University of Nebraska", *Journal of Information Technology Cases and Applications,* 2(1), pp. 41-72.

Silverman, D. (1998), "Qualitative research: meanings or practices?" *Information Systems Journal,* 8(1), pp. 3-20.

Silverman, D. (2000), *Doing Qualitative Research,* Sage Publications, London.

Silverman, D. (2001), *Interpreting Qualitative Data: Methods for Analysing Talk, Text and Interaction,* 2nd edn., Sage Publications, London.

Singh, J. V. (1986), "Performance, Slack, and Risk Taking in Organizational Decision Making", *Academy of Management Journal,* 29(3), pp. 562-585.

Singh, V. (1997), "Systems Integration - Coping with Legacy Systems", *Integrated Manufacturing Systems,* 8(1), pp. 24-28.

Skok, W. and Legge, M. (2001), "Evaluating Enterprise Resource Planning (ERP) Systems Using an Interpretive Approach", in Serva, M. (Ed.) *Proceedings of the ACM SIGCPR Conference on Computer Personnel Research* ACM Press: Sandiego, USA, pp. 189-197.

Smith, N. C. (1989), "The Case Study: A Vital Yet Misunderstood Research Method for Management", in Mansfield, R. (Ed.) *Frontiers of Management*, Routledge, London, pp. 50-64.

Soliman, F. and Youssef, M. A. (1998), "The Role of SAP Software in Business Process Re-engineering", *International Journal of Operations and Production Management,* 18(9), pp. 886-895.

Stake, R. E. (1994), "Case Studies", in Denzin, N. K. and Lincoln, Y. S. (Eds), *Handbook of Qualitative Research*, Sage Publications Inc, Thousand Oaks, pp. 236-247.

Stedman, C. (1999a), "Apparel Makers Anxiously Await SAP Fixes", http://www.computerworld.com, Accessed: 31 May 1999.

Stedman, C. (1999b), "Deal Extends SAP's Warehouse Capabilities", http://www.computerworld.com, Accessed: 9 June 1999.

Stedman, C. (1999c), "Reebok Steps Up Use of SAP Apparel Apps", http://www.computerworld.com, Accessed: 26 July 1999.

Stefanou, C. J. (2001), "A Framework for the Ex-ante Evaluation of ERP Software", *European Journal of Information Systems,* 10(4), pp. 204-215.

Stoecker, R. (1991), "Evaluating and Rethinking the Case Study", *Sociological Review,* 39(1), pp. 88-112.

Stowell, F., West, D. and Stansfield, M. (1997), "Action Research as a Framework for IS Research", in Mingers, J. and Stowell, F. (Eds), *Information Systems: An Emerging Discipline*, The McGraw Hill Companies, London, pp. 159-200.

Strauss, A. and Corbin, J. (1994), "Grounded Theory Methodology", in Denzin, N. K. and Lincoln, Y. S. (Eds), *Handbook of Qualitative Research*, Sage Publications Inc, Thousand Oaks, pp. 273-285.

Sumner, M. (2000), "Risk Factors in Enterprise-wide/ERP Projects", *Journal of Information Technology,* 15(4), pp. 317-327.

Susman, G. I. and Evered, R. D. (1978), "An Assessment of the Scientific Merits of Action Research", *Administrative Science Quarterly,* 23(4), pp. 582-603.

Swanson, E. B. and Beath, C. M. (1989), *Maintaining Information Systems in Organizations,* John Wiley and Sons, Chichester.

Taylor, M., Moynihan, E. and Wood-Harper, A. T. (1997), "Knowledge for Software Maintenance", *Journal of Information Technology,* 12(2), pp. 155-166.

Teale, M., Dispenza, V., Flynn, J. and Currie, D. (2003), *Management Decision Making: Towards an Integrated Approach,* Pearson Education Limited, Harlow.

Themistocleous, M., Irani, Z. and O'Keefe, R. (2001), "ERP and Application Integration", *Business Process Management Journal,* 7(3), pp. 195-204.

Trauth, E. M. and Cole, E. (1992), "The Organizational Interface: A Method for Supporting End Users of Packaged Software", *Management Information Systems Quarterly,* 16(1), pp. 35-53.

Travers, M. (2001), *Qualitative Research Through Case Studies,* Sage Publications Ltd., London.

Truex III, D. P. (2001), "Three Issues Concerning Relevance in IS Research: Epistemology, Audience and Method", *Communications of the Association for Information Systems [http://www.cais.isworld.org],* 6(24), pp. 1-11.

Walsham, G. (1993), *Interpreting Information Systems in Organizations,* John Wiley and Sons, Chichester.

Walsham, G. (1995a), "The Emergence of Interpretivism in IS Research", *Information Systems Research,* 6(4), pp. 376-394.

Walsham, G. (1995b), "Interpretive Case Studies in IS Research: Nature and Method", *European Journal of Information Systems,* 4(2), pp. 74-81.

Warboys, B. C., Greenwood, R. M. and Kawalek, P. (2000), "Modelling the Co-evolution of Business Processes and IT Systems", in Henderson, P. (Ed.) *Systems Engineering for Business Process Change*, Springer, London, pp. 10-23.

Warren, I. (1999), *The Renaissance of Legacy Systems,* Springer, London.

Wastell, D. G., Kawalek, P. and Willetts, M. (2001), "Designing Alignment and Improvising Change: Experiences in the Public Sector Using the Sprint Methodology", in Smithson, S., Cricar, J., Podlogar, M. and Avgerinou, S. (Eds), *Proceedings of the 9th European Conference on Information Systems* University of Maribor: Bled, Slovenia, pp. 1125-1136.

Watson, T. J. (2002), *Organising and Managing Work,* Prentice Hall, London.

Webster, J. and Watson, R. T. (2002), "Analyzing the Past to Prepare for the Future: Writing a Literature Review", *Management Information Systems Quarterly,* 26(2), pp. xiii-xxiii.

Weing, R. P. (1984), "Finding the Right Software Package", *Journal of Information Systems Management,* 8(3), pp. 63-70.

Weinstein, D. and Weinstein, M. A. (1991), "Georg Simmel: Sociological Flâneur Bricoleur", *Theory, Culture and Society,* 18(2), pp. 151-168.

Welke, L. A. (1981), "Buying Software", in Cotterman, W. W., Enger, N. L. and Harold, F. (Eds), *Systems Analysis and Design: A Foundation for the 1980's. An Invitational Conference and Workshop* North Holland: Atlanta, pp. 400-416.

Westrup, C. (2002), "Discourse, Management Fashions and ERP Systems", in Wynn, E. H., Whitley, E. A., Myers, M., D. and De Gross, J. I. (Eds), *Proceedings of the International Federation for Information Processing TC8/WG 8.2 Working Conference on Global and Organizational Discourse About Information Technology* Kluwer Academic Publishers, Barcelona: Spain, pp. 401-418.

Willcocks, L. and Sykes, R. (2000), "The Role of the CIO and IT Function in ERP", *Communications of the Association for Computing Machinery,* 43(4), pp. 32-38.

Wilson, M. and Howcroft, D. (2000), "The Politics of IS Evaluation: A Social Shaping Perspective", in Orlikowski, W. J., Ang, S., Weill, P., Kramar, H. C. and DeGross, J. I. (Eds), *Proceedings of 21st International Conference on Information Systems* Associated for Information Systems, Atlanta: Brisbane, pp. 94-103.

Wilson, M. and Howcroft, D. (2002), "Re-conceptualising Failure: Social Shaping Meets IS Research", *European Journal of Information Systems,* 11(4), pp. 236-250.

Wood-Harper, A. T. (1985), "Research Methods in Information Systems: Using Action Research", in Mumford, E., Hirschheim, R., Fitzgerald, G. and Wood-Harper, A. T. (Eds), *Research Methods in Information Systems: Proceedings of the IFIP WG 8.2 Colloquium* North Holland, Amsterdam.: Manchester, pp. 169-191.

Wood-Harper, A. T., Antill, L. and Avison, D. (1985), *Information Systems Definition: The Multiview Approach,* Blackwell Scientific Publications, Oxford.

Yiannis, G., Fineman, S. and Sims, D. (2000), *Organizing and Organizations,* Sage Publications, London.

Yin, R. (1989), *Case Study Research: Design and Methods,* Sage Publications, London.

Yin, R. (1994), *Case Study Research: Design and Methods,* 2nd edn., Sage Publications, London.

Yin, R. K. (1981), "The Case Study Crisis: Some Answers", *Administrative Science Quarterly,* 26(1), pp. 58-65.

Yin, R. K. and Heald, K. A. (1975), "Using the Case Survey Method to Analyze Policy Studies", *Administrative Science Quarterly,* 20(3), pp. 371-381.

Yourdon, E. (1986), *Managing the Structured Techniques: Strategies for Software Development in the 1990s,* 3rd edn., Yourdon Press, Englewood Cliffs.

Zuboff, S. (1987), *In the Age of the Smart Machine,* Basic Books, New York.