A View into the development of two comparable systems through the application of business re-engineering in an SME

Nattrass, Carl

Carl Nattrass

Submitted for the degree of Master of Science

University of Durham 2007

A View into the Development of Two Comparable Systems through the Application of Business Re-engineering in an SME

Abstract Page

Ideal Caravan Sales Ltd is a medium sized business with their headquarters in Durham, England. Their main business is the sale and transportation of static and touring caravans.

Ideal Caravans (IC) have identified a process within their business which is not functioning adequately and does not allow the required control. As a result of this, they have decided to review that process and implement a computer system in order to rectify the problem.

The company have also taken this opportunity to introduce a structured framework which they can apply to all future computer system developments.

This thesis is an examination into the highlighted problem and discusses options available to the company. It considers off-the-shelf options as well as bespoke systems. The thesis also defines a structured approach for the company to use in the implementation of any future software systems. This will result in positioning the company so that it can clearly identify, design and implement new computer systems successfully whether internally or externally.

The thesis also considers a comparison between two bespoke systems, identifying which is the most successful in solving the identified issue.

Acknowledgments

I wish to acknowledge my thanks to Dr. Liz Burd of the Department of Computer Science of Durham University for her patience and logical reasoning without whom this thesis may have read like a novel.

For her love and understanding, I wish to thank my wife Janice.

For giving me the motivation to climb the mountain, I wish to thank my two sons, Shaun and Daniel.

List of Contents

List of Figures

List of Tables

Declaration

This thesis has not been submitted for a degree at Durham University, or at any other institution. The research conducted in this thesis is the work of the author except where indicated otherwise.

List of Acronyms and Definitions

BPE	Business Process Engineering
BPR	Business Process Reengineering
CGI	Common Gateway Interface
COTS	Commercial off the Shelf (software)
CM	Change Management
CP	Company Procedure
CRM	Customer Relationship Management
GQM	Goal/Question/Metric
GUI	Graphical User Interface
IC	Ideal Caravans
IP	Internet Protocol
ISDN	Integrated Systems Digital Network
IT	Information Technology
KFR	Key Functional Requirements
LAN	Local Area Network
MS	Microsoft
NFR	Non-functional Requirement
OS	Operating System
OSc	Open Source
PC	Personal computer
PDT	Portable Data Terminal
PHP	PHP Hypertext Preprocessor
RHA	Road Haulage Association
SE	Software Engineering
SDK	Software Development Kits
SME	Small to Medium sized Enterprises
TCL	Tool Command Language
TQM	Total Quality Management
TCP/IP	Transfer Control Protocol/Internet Protocol
VB	Microsoft's Visual Basic

1. Introduction

Many large companies have 'Operation Manuals' which provide guidelines to staff and managers on how to go about the day-to-day running of the company. These guidelines will have been tried and evaluated over time, and if adhered to will help to ensure the safe and efficient running of that company.

Periodically, these manuals will be reviewed and updated as a matter of course, but on occasions the companies may feel it necessary to completely re-write them. This might be the case if a company moves premises, changes one of its main products or alters its management structure.

For smaller companies, these guidelines may be vaguer. Some companies may have no guidelines at all, and will rely on word of mouth to pass information on. Others may rely on their staff's abilities to multi-skill and be able to move from one job to another. When companies such as these do identify areas within their business which are without formalised processes, they may not be in a position of having the expertise in order to implement an improved formalised process. This is assuming that they manage to identify the problem domain in the first place.

The objective of this thesis is to provide such a company with a structured approach which it can use when implementing new systems. It will achieve this by examining the manner in which the company currently implements computer systems. A computer system shall be implemented as a part of this exercise.

1.1 Thesis Outline

This thesis involves a company, which has identified a problem within a business process, but is unsure as to how to resolve it. The area of concern is in the organisation of the transportation and siting of caravans. Currently, the company's caravans are being transported without sufficient and accurate information being known by the company. This problem has manifested itself in numerous ways such as; disruption in transportation service, unexpected arrivals and collections of caravans, disputed accounts with both hauliers and customers, arranged transport being rearranged or even cancelled by salespeople not involved. The company had decided that the process would benefit from being computerised.

Previously, the company has outsourced work of this nature to external companies but has never been satisfied with the results. The company has tried numerous system suppliers and feels that they have been let down each time, never getting a system which wholly solved the original problem.

The thesis examines the problematic business process, and goes on to investigate possible solutions to the problem and makes suggestions to the company as to which solutions if any, would be suitable.

The thesis also considers the methods by which the company identifies, appraises and implements new computer systems. Any shortcomings in these methods shall be highlighted in the evaluation chapter and suggestions made which can address these issues. As part of this process, the concept of Business Process Re-engineering (BPR) would be introduced to the company.

1.2 An overview of Business Process Re-engineering

Business process redesign is "the analysis and design of workflow and processes within and between organisations" [Menzies98]. A higher level of efficiency can be achieved through the modernisation of a company's business processes. The two main areas of application of BPR are the people and the processes of the organisation.

Grover et al [Grover95] defined BPR as "the critical analysis and radical redesign of existing business processes to achieve breakthrough improvements in performance measures."

PBR is the key to transforming how people work, as when people work smartly, their efficiency will improve. Even "the act of documenting business processes alone will typically improve organisational efficiency by 10%" [Grover95] is a considerable improvement.

People can be motivated through better organisation of the business processes. The increase organisation can be a result of introducing new methods. Davenport [Davenport93] noted that Total Quality Management (TQM) differed from BPR by its mode of application. TQM was the gradual change of process over time to benefit the organisation, where BPR mainly takes a radical view in changes and are set within a timeframe.

BPR was originally aimed at work-flows within the business environment, and was to be applied to better business practices. As technology improvements have overtaken business practice improvements, BPR has embraced technology as one of its key tools. There are very few BPR projects in existence today which do not in some way include IT as an integral part of it. Because the computer is such a powerful tool in storing and manipulating information, it has become of major importance in BPR.

It is expected that some of these mentioned benefits of BRP will be felt by the company after its introduction.

1.3 Criteria for Success

The Criteria for Success is the measure by which a thesis is judged against.

The following are the Criteria for Success for this thesis:

1. To demonstrate the successful application of BPR against a real-life scenario in order to create a computer system to correct a problem domain.
2. To conduct an informal experimental evaluation of the above process.
3. To create a structured business method of software engineering focused on the requirements of the company and based on the findings of the above.

1.3.1 The Successful Application of Business Process Re-engineering

A company has identified a problem area within its business and wishes to address that issue by way of reproducing the process within a computerised system. This thesis is the result of the application of BPR upon this process.

1.3.2 The Creation of a Structured Software Engineering Process

This work aims to create a structured software engineering process which the company can use in the future. This documentation could be used internally or externally alongside an external company's procedures to make a more accurate appraisal of the company's current situation and its needs.

1.3.3 An informal Experimental Evaluation of the above Process

The thesis needs to evaluate whether the proposed method of re-engineering was a success, furthermore, if shortcoming are found they should be addressed and a suitable corrective measure proposed. It also needed to ascertain whether the development of two different systems opened up opportunities and was beneficial to the process as a whole.

2. Literature Review – Process Improvement

2.1 Introduction

During the past 25 years, software has become an important aspect of most people's lives. The impact of software on society has increased from a minor role to one in which it would be difficult to exist without; from washing machines, telephones and cars, to simply purchasing groceries at the corner store, all are reliant upon computer software at some stage. As a result of this, there has been an explosion in the amount of software needed, and a proliferation of software complexity.

Unfortunately, software applications can be massively complex and there are a multitude of opportunities where the design and development can lead to unexpected and undesired behaviour. For these reasons, researchers and developers have spent increasing time understanding and developing the techniques and tools available to aid software development. One of the key thrusts in this research is the study, evolvement and improvement of the processes associated with the creation of software. The understanding is that there is a direct relationship between the quality of the development process and the quality of the software as a result of that process [Biro98]. Unfortunately, the software industry lags behind other technical industries such as civil engineering despite much research being done in this sphere, and few companies have implemented software improvement methods. There are however several high profile Companies which have successfully implemented software improvement methods such Motorola [Diaz 97] and Raytheon [Haley96].

This chapter aims to address these issues and offers insight into the current state of the art but is not intended to be exhaustive.

2.2 Concepts of Process Improvement

In order to discuss software process improvement, a number of key words and phrases need to be defined.

The noun *process* can be defined as 'a collection of activities that convert a given input to an output of some value' [Hammer93]. In a software engineering environment, a process may be a wider entity than that which is described above; a process can range from a singular data in/data out occurrence such as an event to handle an inputted date, a series of related events, or creation of an entire application. A better definition by the SEI [Zahran98] is 'the set of activities, methods and practices used in the production and evolution of software'.

Figure 2.1 below taken from Sommerville [Sommerville97] illustrates the way in which the process can be seen as the glue which holds together all the other facets of the software engineering process. By using this visualisation, it is possible to see how altering the process will affect other areas such as management and skills.

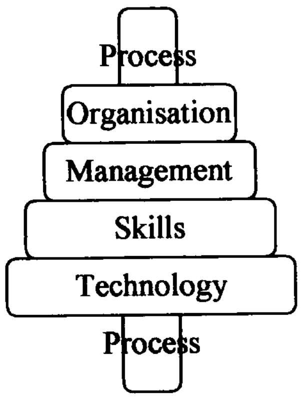


Figure 2.1 The Process as a Key Agent within the Software Engineering Procedure

The *capability* of a process as defined by Paulk [Paulk94] is 'the range of expected results that can be achieved by following a process'. This infers that a process may produce more than one result and is therefore open to deviation by other factors. If a process can be altered to achieve a better range of results then this can be classed as process improvement. A more overall view to process capability is that of the Trillium Experiment [Trillium94] which defines process capability as 'the ability of a development organisation to consistently deliver a product or enhancement that meets customer expectations, with minimal defects, for the lowest life-cycle cost, in the shortest time.'

As a process goes though changes, hopefully changes for the better, it is classed as *maturing*. The concept of maturity was first proposed by the SEI [SEI06] as a way of comparing companies bidding for US Defence contract work. The assumption is that higher the maturity, the better the software system will be. Most of today's software process improvement models utilise this maturity concept. Maturity may come in attained steps; these steps may have criteria connected with them to ratify the reaching of those steps.

In order to measure the improvement in a process it must be possible to assess its state. For this purpose researchers have realised that they can make use of metrics and empirical studies. These are not only to support existing processes but also to evaluate new methods and strategies:

- Metrics can be used as indicators in evaluating concepts such as system complexity, productivity and timescales.
- Empirical methods are needed to guide the evaluation approach of processes.
- Empirical results are the quantative results of applying an empirical method and are useful in comparing the level of success of differing methods.
- Metrics can be used in the assessment of an organisation in deciding its readiness for process improvement.
- The anticipated value to the organisation of through the implementation of process improvement.
- The metrics can also be used in measuring the state of the process improvement implementation.

The Software Engineering Institute [SEI06] states that software *process improvement* is 'an activity that seeks to identify and rectify common causes of poor quality in software systems by making basic changes in the underlying software management process.'

Paulk [Paulk94] preferred the definition 'A description of the stages though which software organisations evolve as they define, implement, measure, control and improve their software processes'. This entirely positive statement is indicative of his researching style which is forward thinking and definitive.

To facilitate PI, according to Sommerville [Sommerville95], an organisation must fulfil a number of basic criteria:

1 It should have a basic cost and scheduling management procedures.
2 It should document all software activities.
3 It should collect measurements of process and software quality.
4 It should operate a continuous process and improvement strategy.

These criteria feature in almost all software process improvement models and form the base on which the software process improvement strategy should be built.

2.3 The Motivations behind Process Improvement

The motivation which lies behind software process improvement can be unique or universal to an organisation; many organisations have benefited from the use of an ISO kite mark or the TickIt Certification on business stationery, others have genuinely benefited from improvements in organisational structure, profit and productivity. Zahran [Zahran98] proposed that these motives may include:

1 The rising costs of software development and maintenance.
2 The need for improved quality of software products.
3 The increasing delays in software projects.

Fuggetta [Fuggetta00] stated that with regard to affecting a company in establishing effective practice, 'there is a complex interrelation of a number of organisational, cultural, technical and economic factors'. Take for example, an organisation that has been told that the only way it will win a contract is if it implements the CMM though-out their development process; what level of motivation might we expect behind its use?

In most cases, companies supply software on the understanding that they make a reasonable profit; where projects have over-run, these profits may be reduced or lost completely. With this in mind, it is a fair assumption that the main motivational factor to a profit-making organisation is to enhance or maintain their profitability. It may be that software process improvement is considered one way of achieving this.

Paulk [Paulk94] stated that the benefits of software process improvement included:

1 The organisation is more responsive to the customer and their market.
2 The lifecycle costs are minimised.
3 The end user's satisfaction is maximised.

Although these are stated in fairly broad terms, the point that he is making is that both the organisation and the customer will benefit from implementations of software process improvement.

It is highly likely that when software process improvement is first discussed within a company boardroom, the most important question will be what the organisation will gain through it. Consider if an organisation can:

1 Achieve a recognised standard.
2 Increase staff retention and motivation.
3 Have shorter development times.
4 Have more reliable software created.
5 Introduce re-useable components.
6 Intercept errors at an earlier stage

All the above factors would benefit the company; the important factor here is the relationship between the benefits and the cost to the company. The main cost to a company would be time; the time taken to learn the software process improvement model and the time taken to implement it.

Another concept is the differing perspective upon motivation behind PI; a software engineer will have different motivations behind an implementation than that of a business manager. The key to a model being accepted by all parties in is its benefits being highlighted successfully. The concept of 'levers' was highlighted by Biro

[Biro98] for just this purpose; he stated that 'levers are means used by a firm to increase its resource generating ability. Resources are used to increase the assets of the firm and reward employees and stockholders'. Although this statement seems slightly lacking in respect of pointing out the benefits to all parties in the organisation, his criteria to successful leverage are quite encompassing; they include financial leverage, operating leverage, production leverage, marketing leverage and human leverage.

The basic motivation behind software process improvement should be that by incorporating a model, an organisation can reduce many of the problems they currently experience though-out a project.

2.4 Obstacles

Where opportunities are available, there is more often than not, opposing obstacles. On the subject of problems encountered with software process improvement, Beecham [Beecham03] stated 'Senior managers cite problems with goals, culture and politics. Project managers are concerned with time scales, change management, budgets and estimates. Developers are experiencing problems with requirements, testing, documentation, communication, tools and technology'. This just about covers everything, but is seen from a worst case scenario. Obstacles are part of the software process improvement paradigm; if it was such an easy concept, models would not be necessary. Beecham [Beecham03] goes on to state that 'documentation, time scales, tools and technology' obstacles are generally related to low maturity companies, whilst 'organisational problems' are more associated with high maturity companies.

Looking at an organisation as a whole, one of the first obstacles to software process improvement is in the attitude of the staff and management towards a new way of working. It may involve new tasks, responsibilities, more paper work or a change of roles. This can be worrying to employees and employers alike. A positive workforce is therefore necessary.

The company may be concerned about implementation costs and the return on investment (ROI). There is also a level of risk involved in software process improvement, for example:

1. Might the implementation of software process improvement slow productivity down initially?
2. What are the chances of failure; if so what are the consequences? Lack of knowledge is a big risk; many organisation are not familiar with best practices, much more emphasis is placed on for example getting training in the latest version of Java rather than on testing or analysis. The chances of failure increases with the level of lack of knowledge.
3. How might the impact of implementing software process improvement into the company effect the rest of the company?
4. How much extra work is involved? Organisations may be pushed for time on deliverables already without the introduction of new methods of working. See the learning curve diagram in Figure 3.2 below. How will staff respond to extra responsibilities; there may be insufficient commitment.

Another obstacle is the confusion caused by having multiple models which all claim to achieve software process improvement. Without sound advice, organisations will struggle to identify the most suitable model. Despite attempts by the likes of ISO, there is also a lack of standardisation amongst how the models are promoted.

The complexity of many software process improvement models may also be a deterrent to many organisations who find the adoption of them to be difficult; CMM recommend an organisation take 18 to 24 months to achieve each level. Humphrey [Humphrey88], founder of the Software Progress Program at the SEI recommends 1 to 3 years per level. A ten-year program of concerted software process improvement may not seem attractive to many organisations.

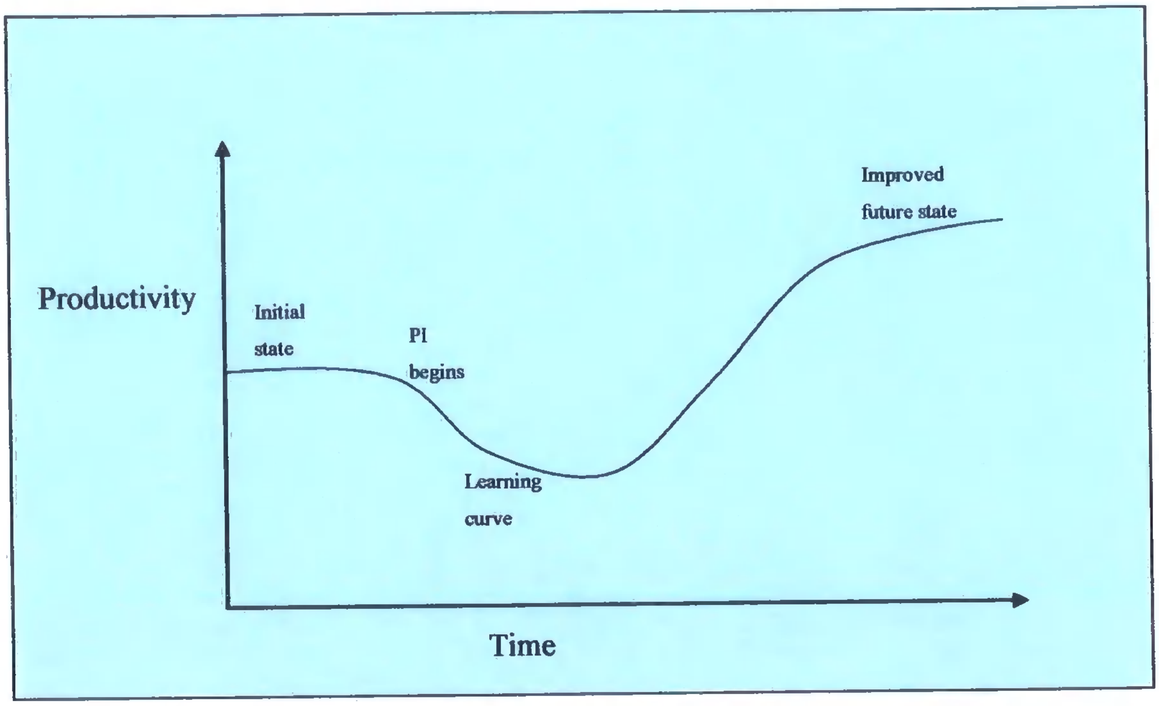


Figure 2.2 The SPI Learning Curve. Drawn from Wieger [Wiegers99]

The benefit of identifying obstacles is that they can be addressed and overcome. Standards such as SPIRE, CMM and ISO have done much research into obstacles; it is in their favour to be proactive in their approach to solving such issues as soon as they are identified.

2.5 Key Researchers

There are a number of key research organisations within the software process improvement sphere. Many work together on joint projects, others may have associations or be completely independent. There are many other organisations and working groups, which have their own theories or enhance existing ones. The following list is not exhaustive but aims to highlight the expanse of research being made currently.

1 SEI: The Software Engineering Institute [SEI06] has taken part in much pioneering work. Their mission is 'to provide leadership in advancing the state-of-the –practice of SOFTWARE engineering to improve the quality of systems that depend on software'. The SEI is affiliated with the Carnegie Mellon University in Pittsburgh, USA. They are involved in developing the CMM.

2 ESI: The European Software Institute [ESI06] is based Bilbao, Spain and is mainly funded by The European Commission. It is currently involved in working on the CMM with the SEI and ISO15504.

3 SPIRAL: Software Process Improvement Research Action Laboratory is a joint venture formed between the University of Oulu of Finland and some key electronic firms such as Nokia. SPIRAL have worked on KNOTS-Q [Saukkonen01] since before 2000. KNOTS-Q is a set of tools and methods for focussing on PI, and its areas include reuse, measurement, software process improvement and multi-site development. The methods are based upon a capability model but differ from the likes of CMM by its inclusion of experimentation into its framework.

4 SEL: The Software Engineering Laboratory [SEI06] is an organisation sponsored by the NASA and was created to investigate the effectiveness of software engineering technologies when applied to the development of applications software. The SEL was created in 1976 and has been in continuous operation, providing software engineering products and performed engineering services as well as conducting a workshop each year since its inception. The SELs aims are to understand the software development process, to measure the effects of various methodologies, tools, and models on this process and to identify and then to apply successful development practices. IT has created its own model to software process improvement called the NASA Approach which focuses on improving the product rather than processes; hence it is a product based approach.

5 SPIRE: The Software Process Improvements in Regions of Europe [SPIRE106] is a European consortium which aims to encourage organisations to implement software process improvement into their business practise. Utilising the CMM up to level 3, they have researched differing methods of its implementation with a view to providing a more simplified path for a non specialised audience to benefit from software process improvement improvements.

6 SPICE: Software Process Improvement and Capability Determination [SPICE06] is an organisation and also the name of a software process improvement model. The first draft of the model was released in 1995

and was adopted as ISO/IEC15504 standard in 1995. SPICE continues to evolve as more research is completed. SPICE is currently one of the most successful research organisations in the software process improvement field today.

7 Microsoft Corporation [Microsoft06]: Due to their influence in the world market, they are in an enviable position to be able to implement software process improvement into their software. Microsoft has incorporated many of the features of CMM into their Visual Studio development suite since 2005. They have also incorporated lean manufacturing; a term used for business performance improvements into many of their Office products. Although not currently working on their own software process improvement model, Microsoft researches aspects such as the effects of GUI design, and development tools on software process improvement.

8 The International Standards Organisation [ISO06]: The ISO occupies a position between the public and private sectors and it defines standards across a wide range of areas. The ISO consider research, propose and approve standards including those of software process improvement.

2.6 History of Software Process Improvement

Fuggetta [Fuggetta00] states that software process improvement has its roots in the development of structured programming languages such as Pascal and C. After this, researches concentrated on the development of design methods and principles such as top-down refinement and information hiding. Finally, there were those researches that put forward software lifecycles such as Royce's [Royce87] Waterfall method. The notion of a piece of software having a lifecycle was new and it was these lifecycles which were most influential in the advancement towards software process improvement.

In 1976, NASA started their own research into software process improvement [Landis92] in conjunction with the University of Maryland. As software was viewed as a key to NASA's success, the Software engineering laboratory (SEL) was set up. Until around 1992 they had worked on integrating SI into a waterfall method based loosely on the Royce's. Since then, there have been many changes, both organisationally and monetary, and much of this work has since been outsourced and the CMM model has principally been engaged. Its aim was to produce manageable, reliable cost effective software. Despite the limitations of the adopted Waterfall method, the SEL produced a structured, clearly defined and good sense approach to an environment which at that time needed clarity and direction. In hindsight the SEL's model was in-depth but cumbersome and would be viable only with large projects; which was exactly what it was designed for. It would have been of little use to many other companies because of this.

In 1989, the ISO and IEC established a committee which set about converting the SOFTWARE ENGINEERING 'cottage industry' of the 1970's into a more formalised process. The resulting standard 12207 was voluntary to organisations. It offered a complete set of processes for acquiring and supplying software. Singh [Singh93] states that it existed through a relationship with ISO9001 and SPICE at that time. He stated that 12207 which was a life cycle process filled the niche between the quality assurance of ISO9001 and the process assessment of SPICE.

Bootstrap was first conceived in 1991 as a European initiative with the aim to speed up the software development processes in Europe. It was an assessment methodology which aimed to highlight where organisations could make improvements. But in 1993, attention was turned to ISO/IEC15504 a worldwide initiative of over 20 countries. It originally was named SPICE. It drew concepts from the CMM and incorporated self-assessments, maturity levels and the documentation process.

During the period of 1988 to present a model called CMM was being developed by the SEI with Carnegie Mellon University. The CMM is still considered to be most popular of the software process improvement models. It is used to determine a company's current process capabilities and to identify the issues most critical to software quality and process improvement. The SEI have also created the Personal Software Process (PSP) which is being taught by universities in the US addresses the improvements needed to be made by software engineers. It covers planning, measurements, and quality control.

Many other groups and organisations have developed spin offs to the major models, some independently, some as part of a larger organisation such as McFeeley's [McFeely96] IDEAL.

2.7 Software Process Improvement Tools and Methods

Software process improvement is not only about the use of tools and methods, improvements can be made by some simple changes in working practices such as better documentation, planning, appraisals of completed projects etc. But when working in such a complex area as software development, it may be beneficial to utilise a tried and tested tool. There are a number of such tools and methods available for software process improvement which are more widely used than others. The following sub-sections highlight the most commonly used ones:

2.7.1 The Capability Maturity Model (CMM)

This model defines the requirements of an ideal company, and offers process management and quality improvement concepts to software maintenance. It is used to determine a company's current process capabilities and to identify the issues most critical to software quality and process improvement. It was initially proposed by Humphrey [Humphrey88] in 1988 and since then has been refined and expanded by the SEI at the Carnegie Mellon University with help from the US government. The model provides 5 levels of process maturity from ad-hoc state to one of continuous process improvement as can be seen in Figure 2.3 below.

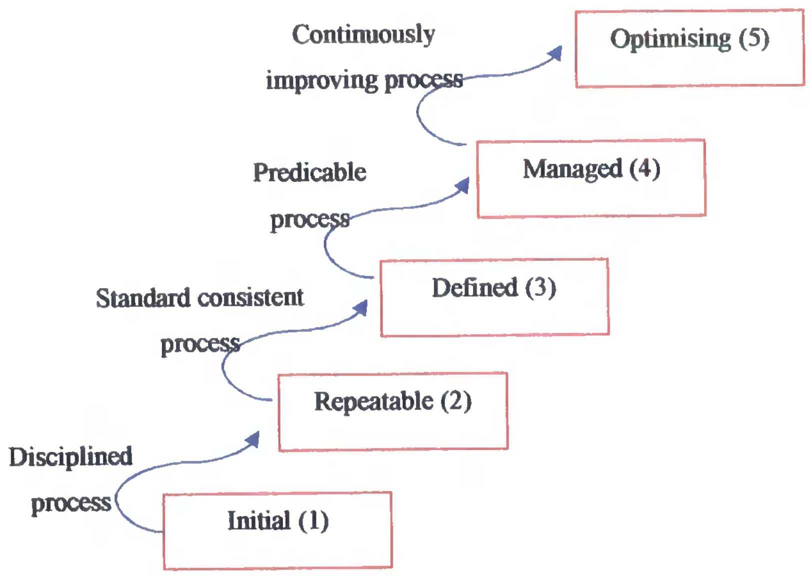


Figure 2.3 The CMM's 5 stages of Maturity [Humphry88]

To achieve each step, the organisation must fulfil certain criteria, control and process. Each level comprises a set of process goals, which as they are satisfied, each stabilise a component within the process. As each level is reached, it must be maintained continuously. To ascertain which level an organisation is on and is capable of, questionnaire and evaluation are used.

2.7.2 ISO/IEC 15504 / SPICE Draft Standard for Software Process Assessment

ISO stands for International Standard Organisation. IEC stands for International Electrotechnical Commission. This standard was originally named SPICE which stood for Software Process Improvement and Capability Determination. SPICE was collaboration of over 20 countries in an effort to create a vehicle to evaluate and improve software processes. It originated in 1993, and in 1997, evolved to ISDO/IEC 15504 which it is currently known. There are traces of the ISO 9001 series within it. There are 3 modes to 15504:

1 The Capability Determination Mode: Determines the capability of a potential software supplier.

2 The Process Improvement Mode: To help improve an organisations own software development processes.

3 The Self Assessment Mode: To help an organisation determine its own ability to undertake a new project.

15504 benefits software purchasers by enabling them to determine the capability of software suppliers and assess the risk of one supplier compared against another. The architecture of 15504 is split into two categories:

1 The Process Dimension. This is a Customer/Supplier level; includes engineering, support, management and organisation activities.

2 The Capability Dimension. The model has 6 levels of capability; each one having a set of attributes which work together to provide a major enhancement in the capability to perform a process.
Each level of capability may or may not involve several defined 'Progress Categories', which must be achieved in order to reach that level. Categories cover areas such as 'how to supply software', 'managing customer's needs' and 'integrating and testing software'.

2.7.3 The Bootstrap Initiative:

This started in 1991 as a European initiative and its early goal was to speed up the application of software engineering technology in Europe. Since 1993 it has been continued as non profit making organisation with member keeping the method up to date.

Bootstrap is an assessment methodology used for determining where an organisation stands in terms of process maturity. By identifying an organisations strength and weaknesses, improvement guidelines can be offered. Essentially, it is only one component in the software improvement model. There are 3 dimensions:

1 Organisation: The rules of management and leadership
2 Methodology: The methods of developing software and their projects
3 Technology: Development tools, process optimisation and automation.

2.7.4 Process Modelling Languages (PMLs):

These are languages and formal specifications which allow the precise and comprehensive representation of software activities, roles and structures. In the past 5 years, increasing emphasis has been placed on the research of PML as part of PI. But whilst PMLs are undoubtedly a valuable tool within software process improvement, they suffer from non-standardisation and many are complex and as a result constitute a large time investment against only a small factor of the software process improvement paradigm. Conradi [Conradi95] states that the key to successful inclusion of a software process improvement is in its standardisation and interoperability as this makes its inclusion into an organisation's software lifecycle easier.

2.7.5 Software Process Improvements in Regions of Europe (SPIRE)

Although not strictly a model or tool, the SPIRE [SPIRE206] organisation offers a more simplified version of the CMM implementation. As one reason why many organisations fail to implement software process improvement is its complexity, this approach may be important in the wider acceptance of software process.

2.7.6 User Groups/Repositories

There are in existence a small number or organisations which offer repositories for software process improvements tools, methods and advice. One such organisation is the Software Engineering Information Repository [SEIR06] based at The Carnegie Mellon University in Pittsburg, USA. The SEIR provides a forum for the contribution and exchange of information concerning process improvement activities. Members can exchange questions or tips and contribute experiences or examples to assist each other with their implementation efforts.

2.7.7 Networks

There are a number of established forums and user groups which encourage the sharing of knowledge within the software process improvement field; one such group is called the Software Process Improvement Network (SPIN), and has members around the world such as the Ottawa SPIN [SPIN06]. Most SPINs are associated with sponsors and partners, and have members from academia and business backgrounds.

2.8 Software Process Improvement Successes

There have been many documented successes of organisations utilising PI. Many of these are published by the organisations that have worked on the model. The importance of these successes lies not only in the results in terms of improvements made, but also in the experiences gained in the operation of the implemented method. Just as software process improvement in an organisation is an iterative process, so too is the evolution of the model.

SPIRE have published a number of papers illustrating the success of their method; one such report focussed on a Company called DataNord in Milan, Italy which noted the following results after application:

1 15% reduction in development time.
2 Integration and debugging time was reduced by 20%
3 Customer satisfaction increased.
4 Personnel confidence and motivation increased.
5 Documentation improved.

The level of software process improvement applied in this case study as well as others researched was relatively shallow focussing on definition of roles and responsibilities, definition of key phases, improvement in documentation and an introduction of a software repository. It is arguable that this level of improvement could have been reached by simply being more organised, that said, an improvement is better than none, especially when the overheads for implementation are so low.

Ferguson [Ferguson99] reports on of the application of the CMM to a company called The Advanced Information System Inc. The company opted to use the PSP model to target its needs:

1 To deliver defect free software and to satisfy customer's increasingly demanding time-to-market goals.
2 To achieve a sustained competitive advantage.
3 To minimise the impact of staff turnover.

The application of PSP went to a much greater depth than was found in the SPIRE example above. The company stated that the goals in terms of software process improvement were:

1 To improve profitability of development projects by meeting cost estimates and schedule commitments with reasonable consistency.
2 To provide a continuing management focus on the progress and visibility of each project.
3 To enable continuous improvement of the dev process through a changed organisational culture through many small incremental improvements.

Ferguson [Ferguson99] reports in depth about the tangible and intangible benefits, for the purpose of this study, only the tangible benefits shall be listed. These where:

1 Scheduling over-run was reduced from 112% to 5%.
2 Effort was reduced from 87% to –4%.
3 Productivity per KLOC increased from 2days to 0.3 days.
4 Defects were reduced from 1 to 0.3 per KLOC.

Another study into the effect of software process improvement was conducted by Herbsleb [Herbsleb94]. He based his research on findings from a broad spectrum of 13 companies. This included companies such as Hewlett Packard, Texas Instruments and Motorola and was based over a period of 6 years. The software process improvement `model used was the CMM and was applied on a large scale with a view to the companies aiming to reach the highest level of maturity within that period. The results were encouraging and the fact that the research was based over so many can only add credence to them. Herbsleb [Herbsleb94] found that on average:

1 The cost of implementing the software process improvement cost from $500 to $2000 per engineer per year.

2 Productivity showed an increase of:

  a 10% to 70% in LOC.

  b 6% to 25% more defects detected earlier.

3 There was a 15% to 32% reduction in development time

4 There was a 10% to 94% increases in quality of software released, relative to initial quality (pre CMM).

5 The business value classed in terms of return on investment (ROI) varied from 4% to 8%.

Hersleb's [Herbsleb94] research showed two interesting points; the cost of software process improvement and an expected ROI. These are important figures when encouraging the use of software process improvement in business. A ROI of only 4% cannot be classed as highly attractive in isolation, but when coupled with the improvements seen elsewhere in the business such as an increase in staff retention, an increase in staff morale and more organised documentation, it begins to look more promising.

The level of increase found with CMM compared to those of the SPIRE case studies seem to indicate that the deeper level of its application, the better the results. This could also indicate the depth of the level of commitment on behalf of the individual organisations; SPIRE organisations may be looking for an easier path for software process improvement, but will accept lesser results.

Other successes brought about by the implementation of software process improvement are advertised by organisation who act as advisors in this field; The SEI, Quality Logic Software Services and Abacus Technology Corporation to name but a few. Both the SEI and Abacus use CMM, where as QLS use ISO15504.

2.9 Summary

Software process improvement first appeared in the mid 1980's and since then much research has been done in this field. That is not to say that there have not been problems. Fuggetta [Fuggetta00] states that the interest in software process improvement research is waning and this is visible through a number of symptoms:

1 Most technologies developed by the research community have not been transferred into business practice.

2 The number of papers on software process improvement presented at conferences is decreasing.

3 There is an increasing feeling that the community is stuck and unable to produce innovative and effective contributions.

Fuggetta may be correct in what he is saying, but it may be that it is not the software process improvement tools which are lacking, but in their integration into industry.

There are a number of key aspects to software process improvement, namely:

1 *Software process improvement must be viable*: The output an organisation receives from its implementation of a software process improvement must exceed its input. Without this simple rule, most organisations will not be tempted to include software process improvement in their business modus operandi.

2 *Software process improvement must be understandable.* Researchers must pitch software process improvement techniques at a level suitable for understanding to organisations.

3 *Software process improvement must be standardised.* With the creation of the ISO, the software industry has a vehicle to push forward standardisation where reviewed versions of models would be backwardly compatible. software process improvement models must be internationally recognised. It would be preferable to have one standardised world-wide software process improvement model which could be evolved through ongoing research gathered from all the software process improvement communities.

4 *Software process improvement must be scaleable.* Organisation must be allowed to choose the degree to which they wish to incorporate a software process

improvement method, and in doing so must equally be able to gain a result from doing so. The SPIRE project is a good example of this.

5 *Software development creation applications must incorporate the standardised software process improvement method into their architecture.* Companies such as Borland and Microsoft are in the ideal position to do this; they should incorporate the standardised software process improvement method into their software design applications. This could include the collation and recording of data, the definition of users' responsibilities and roles and testing guidelines etc.

Emam [Enam98] found that 'the most important factor in distinguishing between success and failure of software process improvement efforts is the extent to which the organisation is focussed in its improvement effort'. This statement could be applied to most aspects of a company not just software process improvement, such as profitability and is generally true. But that does not say that we should discourage organisations from implementing initially a lower level of software process improvement.

In conclusion, the basic motivation behind software process improvement should be that by incorporating a model or method, an organisation can reduce many of the problems they currently experience though-out a project. But it should not be a dogmatic approach, and the real reason for implementing a software process improvement should not be forgotten. Models and methods are there to be picked over; some parts will work, others might not.

3 The Domain

3.1 Introduction

This thesis will focus on the introduction of a new computer system to Ideal Caravan Sales Ltd. To clearly understand what is needed, the current domain needed to be examined. The following section aims to address this.

In most sales organisations, the sales representatives are responsible for an area or region; the benefit of this being that the sales representatives are not competing with each other internally for sales.
Using the motor trade as an example, a sales team may be organised into departments. One department may cover used vehicles, another new vehicles, and a third trade vehicles. IC have organised their sales team in a different manner from this; any representative may sell any caravan onto any park. As a result of this and to a degree, sales representative may be competing against each other for sales. Going on from this, each sales representative works on his or her own and earns a very basic salary before any commission is paid; the majority of their income comes from the sale of caravans. This results in an environment of competition between the sales people. Sometimes competition can be good amongst a team, but in other cases, it may be counter-productive as shall be discovered. These are issues which will be explained in relation to the proposed system.

3.1.1 The Company

Ideal Caravans Sales Ltd has been in business since 1977. It is a small family run business with approximately 30 staff. The business has three directors who are sons and daughters of the company founder.

IC has its main office at Langley Moor in Durham, but also has a large showground in Clifton, Morpeth, and offices in several other locations such as Rothbury, Bamburgh, Barnard Castle and Warkworth. The company can be classed as distributed in as much as 8 of its staff are sales people and may be working from a number of locations, and a further 6 of its staff are mobile fitters who may work anywhere in the North of England.

3.1.2 The Business

IC sell static and touring caravans to the public and to caravan parks. They are classed in the trade as a dealership, and their area of business covers most of the North of England. Recently, the company has expanded into park ownership. They also offer an established maintenance and caravan transportation service. It is worth noting that the transportation service is arranged though a selection of specialist hauliers. These hauliers are mainly located around Hull which is where most holiday homes are built for the UK market.

3.1.3 The Ideal Caravan's Computer Network

The office at Durham has three servers operating an internal network as well as managing an external private network. The Morpeth office is connected to this Extranet. All other offices which Ideal operates from are fitted with ISDN or Dial-up connections and are allowed ad-hoc access to the Intranet. The Intranet may be regarded as the company's private Internet.

The company believe that through the application of technology, it can gain an advantage against its competitors. It currently has several bespoke systems to assist in the running of the company.

3.1.4 The Problem

The Directors at IC identified a problem within the company's operations which they felt could be addressed with the introduction of a new company procedure (CP). The identified problem was in the lack of control the company and its employees were maintaining over the movement of its own and third party caravans. The lack of control had been manifesting itself in various forms such as:

1. Invoices being unaccounted for and disputed by hauliers and customers
2. Caravan deliveries and collections taking place without sufficient organisation
3. Transportation being rearranged or cancelled by sales persons on an ad-hoc basis
4. Insufficient notice given to other departments within the company to react to the movement of vans.

The Durham showground is small and congested. This leads to occasional logistic problems of getting vans in and out. It is worth noting that when a large caravan comes into Durham, there may need to be some reorganising of stock prior to its arrival and therefore planning of van movements is essential. Further to this, when a number of vans are being collected on the same day, they may need to be positioned in the correct order for collection.

IC requested that an investigation take place into the relating current processes, and that a computerised solution be proposed which would correct any shortcomings. The company management discussed at length the preliminaries of a new computer system, but could not agree on the format of technology to be utilised.

As a result of this, it was suggested that two solutions potentially existed; a Windows based system and an Internet based system. The more suitable of the two would be selected after being evaluated and reviewed.

3.2 A Description of the Current Domain

Analysis of a domain can be described as 'the process of capturing and representing information about applications in a domain, specifically common characteristics and reasons for variability' [Malhotra00]. In this case, the process would be the capturing of data, data flows, roles and processes relating to the transportation and siting of caravans. There are many factors to consider with regard to the options a system developer has in re-engineering a business process. As the thesis was to be based around the creation and comparison of two functionally similar systems using two different operating systems (OS), the numbers of factors were further increased.

The process of analysing the domain and the actual domain would be similar for each system, but the two implementations would need to overcome different difficulties.
It is worth noting that at the time the research was done into the problem domain, IC, despite having over 50 written CP's, had no written procedure for the sale or transportation of caravans. As this is a key area of the company's business, this seemed to be quite an oversight.

3.2.1 Staff

The average age of the company staff was in the 50's and although most members of staff were allocated a PC, there was very little inclination or motivation to learn new skills. The sales team who would be the target users could be classed as being to the lower end of the PC literacy scale. This was an important point, and as most of the sales team were nearing retirement age, some of them may have problems learning new technologies.

IC currently rely on the sales team supplying information to each other by word of mouth to ensure that caravans are transported successfully. The sales manager is the central focal point for communications within the department, and he supplies the rest of the company with printed (or electronic) reports keeping employees up to date.

Figure 3.1 shows a hierarchical view of the company staff who are involved in the process of selling of caravans and their subsequent transport.

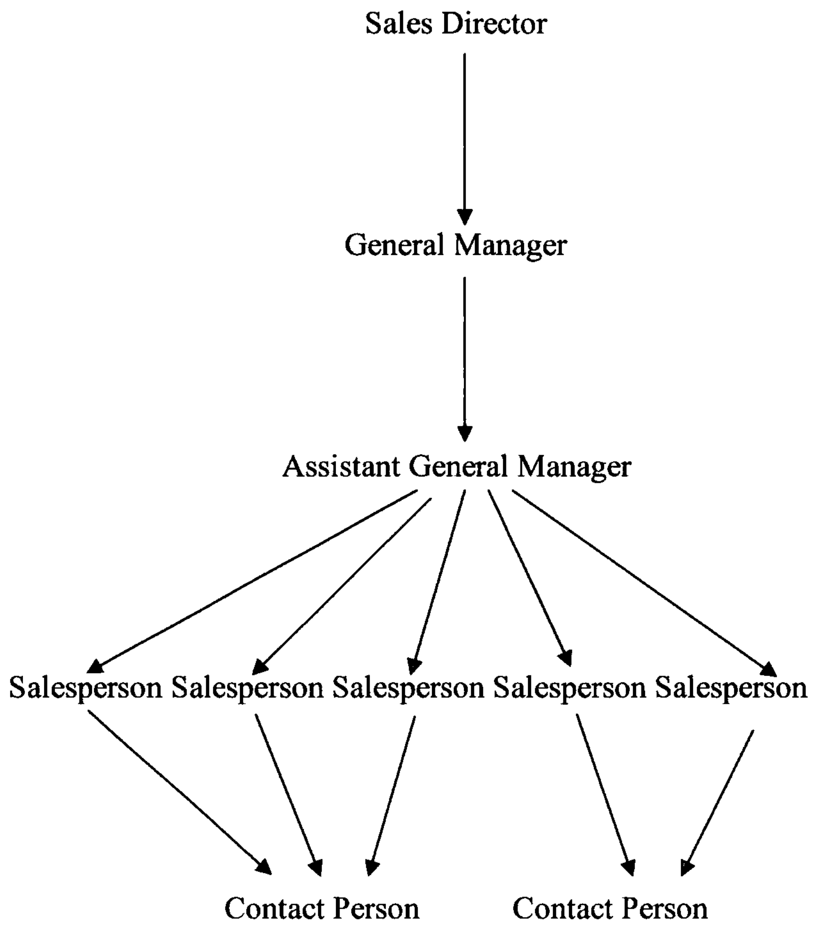


Figure 3.1 Hierarchy of staff involved in caravan sales

As can be seen from Figure 3.1, the company has a five layered hierarchy of staffing within the sales department. Those directly involved in the process of selling caravans extends from the contact personnel who are purely employed to meet and greet customers coming onto the display areas, up to the sales director who is responsible for company wide decisions relating to the sales department.

3.2.2 The Use of Computers within the Company

Emails are increasingly being used in communicating externally to customers and other businesses. Another form of communication used by the company is by their implementation of a centralised database running an application called Maximizer. This is a Customer Relationship Management (CRM) program which allows any information relating to each client to be stored and available to all other users. This in itself is a good communicator as any employee can see any notes which have been entered against a customer's record during any period. This is useful if a sales person has returned after a day off, he or she can see all notes placed on the system by other employees the previous day.

The company has several bespoke systems already in place which have been designed and installed by various contracted companies. None of them had been a great success. It was arguable whether these systems were not a success due to the company being employed to write them or due to IC not specifying sufficiently clearly what was required. One key entity of these systems was a database commonly known as the 'Ideal' database. This held the following information:

1. Authorised users details
2. Caravan parks details (those parks which IC operated on)
3. Lists of caravan manufacturers and their models
4. Lists of statuses. Some entities within the company have a status associated with them. For example:
    a. Vans can be classed as being; on stock, sold, or sold and salesperson paid etc.
    b. Parks are classified according to who may have dealing with them; IC only, anyone dealer, or competitor only etc.
    c. Customers may be classified as being current, buy-in, prospect or deal etc.
5. Marketing lists. There were a number of linked lists such as advertising details e.g. publication->newspaper->Evening Chronicle.

This database must be incorporated into any new systems, so that duplication of data was not necessary.

3.2.3 Company Procedures

As previously mentioned, the company had no formalised procedures in place for the transportation of vans. That is not to say that processes were not complex. The normal process for the arrangement of the transportation of a van is as follows:

1. A customer purchases a caravan.
2. The park is notified of the sale (To organise the tidying of the pitch etc).
3. The park stipulates the date which the siting of van can take place (this date is given around 3 weeks prior to the siting date).
4. The sales representatives check the diary to see when fitters are available (If needed). IC have 3 sets of fitters – each set can be pre-booked when they are known to be needed.
5. The sales representative contacts the hauliers to see who can transport the van at that time.
6. The haulier is arranged. A confirmed entry is made in diary. The park is notified (approx. 2 weeks prior to the siting date).
7. The company Service Manager is provisionally notified of the basic details (approx. 1 week prior to the siting date).
8. The Service Manager is given full details on the week of the movement; such as the number of Land Rovers, any plumbing connection, any gas connection and any servicing needed. It would also include any extras organised such as steps, fridge, gas bottles, veranda etc. and/or a demonstration of the heating system.
9. The collection takes place.
10. The transportation takes place.
11. The siting takes place.
12. The Service Manager confirms the siting completion (that evening).
13. The customer is informed the next day.

The above process provided the sales staff with at most, the following:

1 A basic guideline from which the sales team could work.

2 A diary or diaries in which to store movement details.

3 Basic access to information where physically possible.

4 A backup facility for the information in as much as there are two copies of the diary kept.

5 The distribution of the current week's information via email to certain staff.

6 Minimal security from general public as the diaries are left in public areas.
7 Minimal security from internal people for similar reasons.

The benefits as listed above could only be relied upon as far as the discipline of the staff is concerned. This, as has been illustrated, was not sufficient.

3.2.4 The Recording of the Bookings

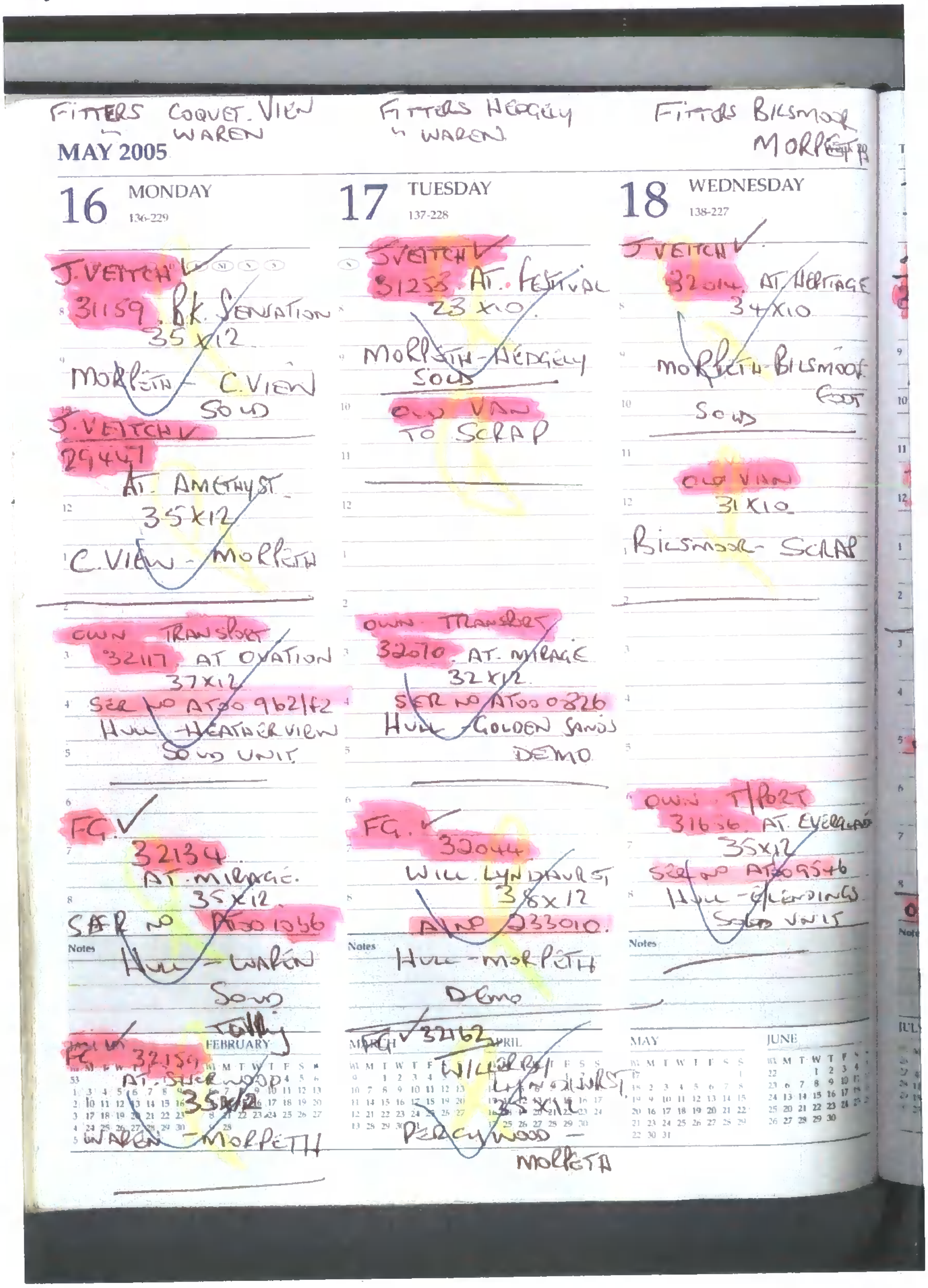
IC currently record the movement of their caravans in two diaries. The master diary is kept at their head office in Durham. There is also a second diary at their Morpeth showground. The master diary contains details of all movement of caravans, where as the Morpeth diary is concerned only with caravans going out of and coming into Morpeth.

It is feasible for a salesperson at Morpeth to sell and arrange transport for a caravan situated at Durham. It is therefore the sales person's responsibility to notify Durham of the details and for them to update their diary. There are occasions when this does not happen, or when changes have occurred and Durham has not been notified, and as mentioned earlier, this can cause problems at the time of the movement.
Sales people arrange transport through the two transport representatives. These transport representatives are responsible for keeping the diary up to date. Unfortunately, as these transport representatives are also sales people, there is a regular case of bias within priorities of bookings.
Both diaries are to contain the same information about the transport arrangement:

1 The date of movement.
2 The haulier.
3 The caravan make and model.
4 The caravan size.
5 The caravan stock number.
6 The caravan serial number – where known.
7 Transport 'From' and 'To' locations.
8 The sale status of the caravan, i.e. sold, trade scrap or stock.

Figure 3.2 Booking Diary Information

The following Figures 3.3 and 3.4 show examples of the Durham booking diary.



FITTERS COQUET VIEW WAREN

FITTERS HEDGELY WAREN

FITTERS BILSMOOR MORPETH

MAY 2005

16 MONDAY 136-229

J. VEITCH ✓
31159 BK. SENSATION
35 X 12
MORPETH – C. VIEW
SOLD

J. VEITCH ✓
29447
AT AMETHYST
35 X 12
C. VIEW – MORPETH

OWN TRANSPORT
32117 AT OVATION
37 X 12
SER NO AT00 962/F2
HULL – HEATHER VIEW
SOLD UNIT

FG ✓
32134
AT MIRAGE
35 X 12
SER NO AT00 1356
HULL – WAREN
SOLD
TALLY

FG ✓
32159
AT SHERWOOD
35 X 12
WAREN – MORPETH

17 TUESDAY 137-228

J. VEITCH ✓
31255 AT FESTIVAL
23 X 10
MORPETH – HEDGELY
SOLD

OWN VAN
TO SCRAP

OWN TRANSPORT
32070 AT MIRAGE
32 X 12
SER NO AT00 0826
HULL – GOLDEN SANDS
DEMO

FG ✓
32044
WILL LYNDHURST
38 X 12
A NO 233010
HULL – MORPETH
DEMO

FG ✓ 32162
WILLERBY LYNDHURST
35 X 12
PERCYWOOD – MORPETH

18 WEDNESDAY 138-227

J. VEITCH ✓
32014 AT HERITAGE
34 X 10
MORPETH – BILSMOOR FOOT
SOLD

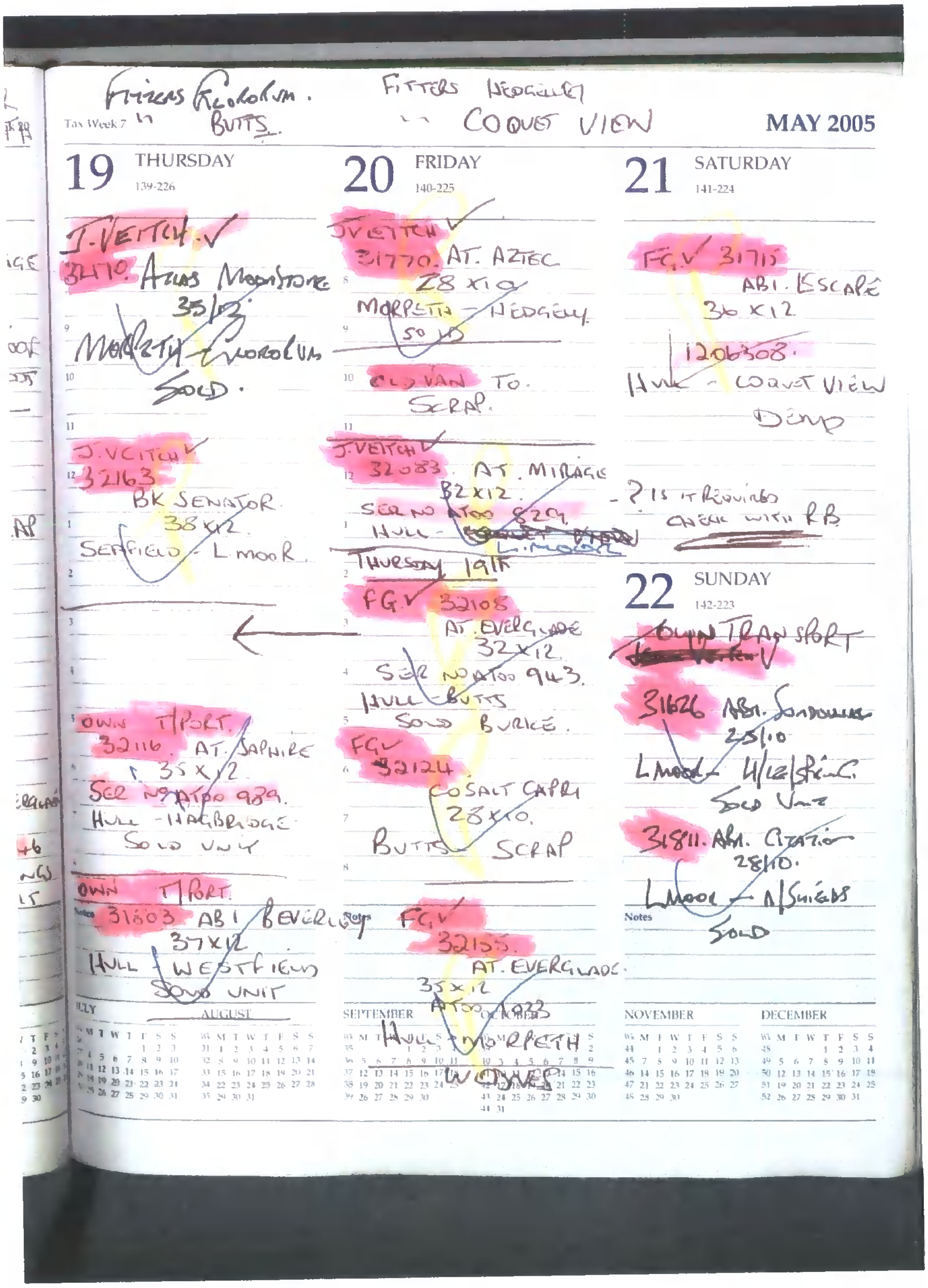
OLD VAN
31 X 10
BILSMOOR – SCRAP

OWN T/PORT
31636 AT EVERGLADES
35 X 12
SER NO AT00 9546
HULL – GLENDINGS
SOLD UNIT

Notes

FEBRUARY MARCH APRIL MAY JUNE

Figure 3.3 The Booking Diary Left Hand Side



Fitters Flororum. Fitters Hedgeley
Tax Week 7 " Butts. " Coquet View

MAY 2005

19 THURSDAY 139-226

J. Veitch ✓
31470. Atlas Maidstone 35/12.
Morpeth - Floral Lum
Sold.

J. Veitch ✓
32163 BK Senator 38x12.
Serfield - L. Moor.

Own T/Port.
32116. At. Saphire 35x12
Ser No ATOO 989.
Hull - Hagbridge
Sold Unit

Own T/Port.
31803 ABI Beverley 37x12.
Hull - Westfield
Sold Unit

20 FRIDAY 140-225

J Veitch ✓
31770. At. Aztec 28x10
Morpeth - Hedgeley
Sold

Old Van To. Scrap.

J. Veitch ✓
32083. At. Mirage 32x12.
Ser No ATOO 829.
Hull - ~~[illegible]~~ L. Moor

Thursday 19th

FG ✓ 32108
At. Everglade 32x12.
Ser No ATOO 943.
Hull - Butts
Sold Burke.

FG ✓
32124
Cosalt Capri 28x10.
Butts Scrap

FG ✓
32155.
At. Everglade
35x12
ATOO 1023
Hull - Morpeth
Wooler

21 SATURDAY 141-224

FG ✓ 31715
ABI. Escape 36x12
12063308.
Hull - Coquet View
Demp

- ? Is it Required
Check with RB

22 SUNDAY 142-223

~~Own Transport~~
~~[illegible]~~ ✓

31626 ABI. Sandowne 25/10
L Moor - H/lee/Spring.
Sold Unit

31811. ABI. Crestation 28/10.
L Moor - N/Shields
Sold

Notes

AUGUST
Wk M T W T F S S
31 1 2 3 4 5 6 7
32 8 9 10 11 12 13 14
33 15 16 17 18 19 20 21
34 22 23 24 25 26 27 28
35 29 30 31

SEPTEMBER
Wk M T W T F S S
35 1 2 3 4
36 5 6 7 8 9 10 11
37 12 13 14 15 16 17 18
38 19 20 21 22 23 24 25
39 26 27 28 29 30

OCTOBER
Wk M T W T F S S
39 1 2
40 3 4 5 6 7 8 9
41 10 11 12 13 14 15 16
42 17 18 19 20 21 22 23
43 24 25 26 27 28 29 30
44 31

NOVEMBER
Wk M T W T F S S
44 1 2 3 4 5 6
45 7 8 9 10 11 12 13
46 14 15 16 17 18 19 20
47 21 22 23 24 25 26 27
48 28 29 30

DECEMBER
Wk M T W T F S S
48 1 2 3 4
49 5 6 7 8 9 10 11
50 12 13 14 15 16 17 18
51 19 20 21 22 23 24 25
52 26 27 28 29 30 31

Figure 3.4 The Booking Diary Right Hand Side

As can be seen from figures 3.3 and 3.4, the details of the transportation are placed in a column representing all of that day's work. The order that they are entered onto the diary is the order in which the caravans were sold; there is no concept of time. This is an important point and often causes confusion in the depots.

A tick next to the haulier is used to confirm the haulier's acceptance of the movement. A tick over the complete entry confirms the movement's completion.

At the top of each day's entry can be seen some hand written park locations. These are used to pre-book a sets of fitters in advance in order to advise users of the diary, and the service department when the fitters may be needed. It also helps stop the fitters from being over-booked.

Both diaries are available for viewing at any time in the respective sales offices. Those offices being at Durham and at Morpeth and are kept up to date by email and by word of mouth periodically, and when entries were made or altered during the week. Photocopies are made of both diaries on a Sunday night and are distributed via internal mail to the staff as listed in Figure 3.5.

1 Sales director – Used for his own information.
2 General manager – Used for his own information.
3 Service manager – Used to organise the teams of fitters.
4 Receptionist – Used to notify the caravan owners on the day of transportation as to its status of completion.
5 Accounts – To check against invoices received from the haulage company.

Figure 3.5 Staff who have need to use the Diary

The two diaries would have their final synchronisation on a Sunday. After that the photocopying and distribution of the pages would take place.

3.3 An Assessment of the Current Domain

Having assessed the domain, it appeared that the lack of fundamental organisation going into the transportation procedure was creating the majority of the problems:

1. The salespeople were not making sure that they were in possession of all the relevant information prior to committing to bookings.
2. The above information was not being recorded correctly.
3. The booking information was not being relayed to other parties who should have been informed.
4. The staff hierarchy was making the process of bookings easier for some employees than others.
5. That same hierarchy was allowing certain employees the right to over-ride other employee's bookings.

Figure 3.6 Problems identified with the Current Process

These were fairly fundamental issues and lent themselves to being implemented in a computer system by their very nature. It appeared that to solve these problems was a case of recreating the process but enforcing a number of set points as can be seen in Figure 3.7.

1. Ensuring the sales person had knowledge of all the relevant transportation information before allowing a booking.
2. Ensuring that that information was recorded in an appropriate manner.
3. Ensuring that all employees involved in the booking process had equal rights in the process.
4. Ensuring that that information was made available to authorised employees when it was needed.
5. Ensuring the integrity of the information.

Figure 3.7 List of Corrective Issues to be addressed by the New System

3.4 What was required

One of the main problems when working with small and medium sized companies is their ad-hoc method of running a business. IC had two main domain problems; the first was its lack of company procedure, the second was its lack of control over the domain. Once this had been identified, it was possible to look for a solution.

Considerable amounts of resources have been deployed by many organisations with a view to correcting such issues as these. The introduction of Total Quality Management (TQM) in the 1950's highlighted the opportunity of examining working practices. This has been expanded numerous times by various researches including the identification of the four main elements within the concept; employee involvement, customer focus, benchmarking and continuous improvement [Daft99]. TQM 'encompasses the total organisation, regardless of a particular discipline' [Anderson99]. The process lends itself to being broken down into process chunks. IC had identified one of these chunks on their own accord.

One possibility considered early on was the use of Microsoft Outlook as an organisational tool, to be used together with its diary facility. Bookings could be entered by any individual in the company and visible to all others. All employees at Ideal already had this software installed and set up on their computers so it would have been easy to implement. Unfortunately, the introduction of a system as such was considered not sufficient as it did not force the users through a course of actions; it did not ensure that the correct information was gathered. Referring to 3.2.3 where it was noted that the discipline of the staff was not sufficient to engage in a system which relied on the staff to oversee themselves. This was more a discipline problem as opposed to a lack of an available tool. The system needed to force users through a set routine in order that the information could be collected accurately and distributed correctly.

One of the key points with the system was the formality aspect. The booking of a van movement should only be allowed when a minimum amount of information was available, without this information, the booking would not be allowed to be processed.

As the information was input, some of it would need to be parsed. These can be seen in Figure 3.8.

1. Dates should be checked against an input mask i.e. dd/mm/yy.
2. Dates should be validated; no booking allowed prior to current date, no more than 3 months in advance etc.
3. Only parks listed on the central database would be allowed to be used. Any new parks would need to be added together with their relevant details.
4. Double bookings should not be possible.
5. Numeric field should not accept Alpha data.

Figure 3.8 Data Entry Parsing

Going on from this, the necessary data needed in order that a booking be made successfully must included an agreement made upon the cost to IC for the transportation of the van and a booking reference be assigned. This could be useful at a later date in the possible automatic creation of cost expectations. This had been a standard procedure within the business for years, and during that period the sales staff had become so lackadaisical in their attitude to this that this cost was being agreed at the point that the invoice was being submitted to IC. This process had to be enforced, and so would become part of the data input procedure.

As has been mentioned, the information kept about the bookings had to be made available, albeit only internally. The possibility of the information being made available over the Internet meant that the data had to be secure. The level of security should be relative to the sensitivity of the data; bank details are more useful than caravan prices. It was sufficient at this stage to acknowledge that only authorised users should have access to the information. Authorised users are those as listed on the IC central database.

Another aspect of security was with the integrity of the data as stated in point 5 of Figure 3.7. Part of data integrity is to ensure that it cannot be spoiled accidentally, or otherwise; only the user who added the information should be allowed to edit or delete it. This would avoid the possibility of bookings being mislaid. It also offered the level playing field scenario which was described in point 4. This was intended to allow any member of sales staff to be able to make any booking without it being passed by more senior members of the sales team, with the possibility of it being reduced in priority.

Another key point relevant to the above was to place the onus of the correctness of the booking information upon the sales person involved. Should a haulier arrive at Morpeth to collect a van and find it located second in the queue in the collection area, either the haulier has arrived early and would have to wait, or the sales person had entered the collection time incorrectly.

The development would create a problem in the maintaining of concurrency over a network, the method by which this would be done would be decided later. At this point it was only necessary to know that the network was not only in one location but could be further expanded as the company developed.

3.5 Key functional requirements

To identify requirements, the developer needs to be able to conceptualise the system in terms of functions, processes, inputs and outputs [Mylopoulos00]. Requirements may be viewed in part as a wish list for functionality, and partly as those processes needed integrally to the overall system. In order for this computer system to address those issues discussed above, it would need to include certain functionality. It is possible to itemise those functions having examined the current domain.

Table 3.1 will be used as a guide to the key functional requirements (KFR's) needed in order to achieve a minimum required system. It is a remodeling of the current system with initial expectations of what a computerised system would provide. Additional functionality may be considered during the development process.

The letters 'R' and 'D' in the heading refer to Required functionality and Desired functionality.

Req. No.	Feature	Applies to	Functionality	R	D
1.1	Data entry	Hauliers details	Allow add, edit and delete	Y	
1.2.1		Bookings	Allow add, edit and delete	Y	
1.2.2			Check for double bookings	Y	
1.2.3			Record information relating to the siting of the caravan once it has been delivered.	Y	
1.2.4			Record information relating to the transport arrangements.		Y
2.1	Security	Users	Password protection	Y	
2.1.1			Password should be checked against IC main database	Y	
2.2		Paper-trail	A log to record changes to the system data		Y
3.1	GUI	Diary screen	This should look like a diary. It should have daily and weekly views.	Y	
4.1	GUI	Diary screen	Allow users to pre-book fitters and hauliers		Y
5.1	GUI	Data entry form	Needed clarity and simplicity.	Y	
5.1.1			Direct the user though key steps of booking.	Y	
6.1	GUI	General	Should be designed with the company's Corporate Image in mind	Y	
7.1	Distribution	Access to data	Must be accessible at all company offices.	Y	
8.1	Compatibil-ity	Existing data	Use company's database for park names	Y	
8.1.1			Use company's database for passwords	Y	
8.2.1		Application	Must be compatible with existing network	Y	
9.1	Reports	Hauliers	Showing reliability and accident rate		Y
9.2		Users	Showing credibility of bookings, number of incomplete jobs etc		Y
10.1	Data Availability	Entire System	The data should be available either offline or online.		Y

Table 3.1 Key Functional Requirements

4. Ideal Caravans Business Process Engineering – A Case Study

4.1 Introduction

Chapter 2 highlighted a process within Ideal Caravan's operations, which was causing concern to the company. The problem lay in the lack of control the company and its employees were maintaining over the movement of its own and third party caravans. This chapter offers a study into the BPR of these identified issues.

There were two options identified to solve this problem:

1 Utilising a currently available package or a commercially available off the shelf package (COTS)

2 Creating a bespoke system.

Option 1 would be considered first as it offered the quickest solution; if a suitable piece of software could be found it would be possible to have it installed and running in a far shorter time period than Option 2. There was also a good chance that the COTS package would be considerably cheaper than Option 2. There would be a trial of any currently available suitable systems and a judgment made of their suitability. In order to do this, a set of evaluation criteria needed to be created to be used as a benchmark comparison.

4.2 The Existing Process

Figure 2.2 listed the steps that the staff should be taking to successfully complete the process under examination. Unfortunately, much of the time these steps were not being completed; in some cases, no documentation of any kind was in existence. Much of the organising was being done over the telephone and any notes made were being placed into personal diaries or onto scraps of paper. As there was no formal CP in place, and there had been no long-term study into how the individuals in the sales team perceived this process, it was difficult to establish exactly how organised or disorganised they actually were. During the project, once the sales teams' working practices began to be examined, they started paying more attention to them which was misleading in as much as a true picture of the working practices was not being observed.

4.3 The Proposed Process

The computerised system should at minimum be able to cater with the collection of data relating to those steps in Figure 2.2. More realistically, it should guide the users through a series of steps collecting the necessary data as part of that process. Those steps listed appeared initially to be adequate as a basis for recreation by engineering, but there were areas such as communication between sales representative, park owner, haulier and the Service Department which needed improving.
The proposed process should also be considered alongside the requirements as listed in Table 2.1.

4.4 The Evaluation criteria

One of the problems facing system designers/evaluators is how to effectively evaluate system design to support multiple interactive users. Users are different; they come with differing viewpoints, expectations and skills. Ross [Ross95] suggested a method of 'combining theoretical analyses and participatory design methods'. Further to this, she explained how with her PETRA Framework, the 'continued involvement of users and full active participation in the evaluation process' was integral with its success. Although, the PETRA framework was not to be implemented here due to it being based upon collaborative work and would have involved unnecessary complications, an attempt was made to involve the potential system users as much as possible.
QOC [McKerlie93] was identified as a suitable method to illustrate the evaluation discussion.

The evaluation of software is not purely to check suitability, but can also be used as an opportunity to see how other organisations are utilising systems in order to organise their own procedures and working practices.

The company supplied a list of criteria which should be considered whilst testing the systems. This list was reviewed and after consideration and discussion, the criteria was agreed and is found in Table 4.1 below.

Question	Which of the available systems, if any, would be most suitable for IC
Option 1	RHA Roadrunner
Option 2	BookingSystem.com
Option 3	MyBooking.com
Option 4	TrackWhere
Criteria 1	**Cost**: Although the implication of cost is not paramount, it is still relative. If comparing two systems, one costing £50 per user, the second costing £400 per user, it would be expected that the latter system should offer more than the former.
Criteria 2	**Data availability**: The accessibility of the data was integral to the system's success; would the system offer connection via the company's internal network, its external network, or could it be made available from any location via an Internet connection? The system would need to serve at least 10 users.
Criteria 3	**Expandability**: Needs to take into account the possibility of expansion or alteration of the system. If the system reviewed does not closely match what Ideal Caravans were after, might it be possible to have the vendor make changes to it?
Criteria 4	**Key Functional Requirements**: This is the list of requirements found in Table 2.1. It covers system features, functionality and appearance.

Table 4.1 Using QOC to illustrate The Systems and Required Criteria

4.5 The Evaluation

When the company decided to introduce a new system to control transport bookings, one of the first considerations to be taken into account was whether there already existed an application which would be suitable. As mentioned previously, Microsoft Outlook had first been considered. This idea was discarded as it relied too much on the user supplying the correct information. What was needed was a system which enforced the need to collect and document the correct information before a booking be made possible. Investigation showed that there were quite a number of booking systems available, some of which compared to creating a bespoke system were very cost effective. The basic fundaments of a booking system are quite common throughout many business areas. This increased the chance that a viable package might be found to suit the company's needs.

Initially, The Road Haulage Association (RHA) was contacted for information. This association is a non-government body that specialises in advising on all issues relating to road haulage. They are, relatively speaking, what the British Computer Society is to Information Technology (IT) professionals. The RHA had made it known that they have developed their own systems under the name of Roadrunner, which would be reviewed. There are two versions; one a stand-alone application, the second, accessed via the Internet. Both applications were similar in functionality other than their method of distribution and number of users.

Further to this, some more generic booking systems which were available were reviewed such as BookingSystem.com and YourBooking.com. These are suitable for most small to medium sized businesses and can be altered to suit the individual business. An important aspect to these would be how much alteration would be available and would it be sufficient for what was needed. The problem might arise that the company was not just looking for a system which managed the transportation of caravans, but also managed what their fitters were to do with the van once it was delivered.

Also reviewed were some packages such as Trackwhere which are produced by software companies specialising in haulage organisational software, and also a suite specially designed for large multinational companies.
Finally, the possibility of IC employing a software development company to produce a bespoke system was considered.

4.5.1 R.H.A. Roadrunner

The RHA has worked in partnership with a number of software developers, chiefly RoadTech, to create a package called RHA Roadrunner for Windows. RoadTech have produced a number of similar systems in the past. Their first version was distributed around 15 years ago. Information about RHA Roadrunner can be found at www.roadrunner.uk.com/products/transx.shtml.

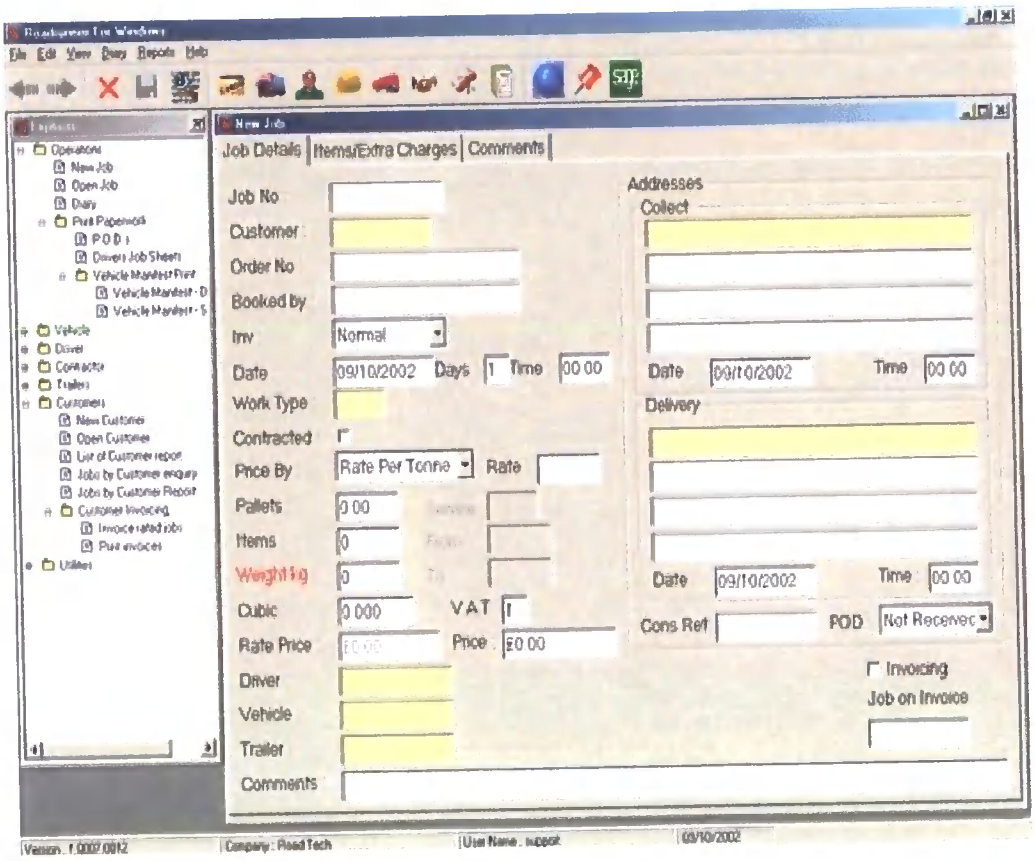
Cost
This package retails for £100 and this includes a single user license. Each further license cost £100 thereafter.

Data Availability
The system would be distributed over an internal network with the option of a dedicated machine to host the database. There would be no offline capability

Expandability
RoadTech had stated that they were not in a position to make any changes to any of the field descriptions, which would have been useful to IC.
The main data entry screen, Figure 4.1 offered a third tabbed area providing a large text box which could have been used for recording additional information unique to IC, but unfortunately no searches or data extraction could have been applied to this data. Requirement 9.1 of Figure 2.1 which listed functional requirements states that it would be desirable to be able to create a number of reports based on some of this data, for example to illustrate the reliability and punctuality of hauliers. This would not be possible in this case.

Functionality/Interface

The application is negotiated by using an Explorer style list down the left hand side of the screen and a group of menus along the top. Vehicles and driver details need to be set up and then jobs can be created. The application caters for contract work and for different types of haulage trailers.



Job No	
Customer	
Order No	
Booked by	
Inv	Normal
Date	09/10/2002
Days	1
Time	00:00
Work Type	
Contracted	
Price By	Rate Per Tonne
Rate	
Pallets	0.00
Items	0
Weight kg	0
Cubic	0.000
VAT	1
Rate Price	£0.00
Price	£0.00
Driver	
Vehicle	
Trailer	
Comments	

Addresses

Collect	
Date	09/10/2002
Time	00:00
Delivery	
Date	09/10/2002
Time	00:00
Cons Ref	
POD	Not Received
Invoicing	
Job on Invoice	

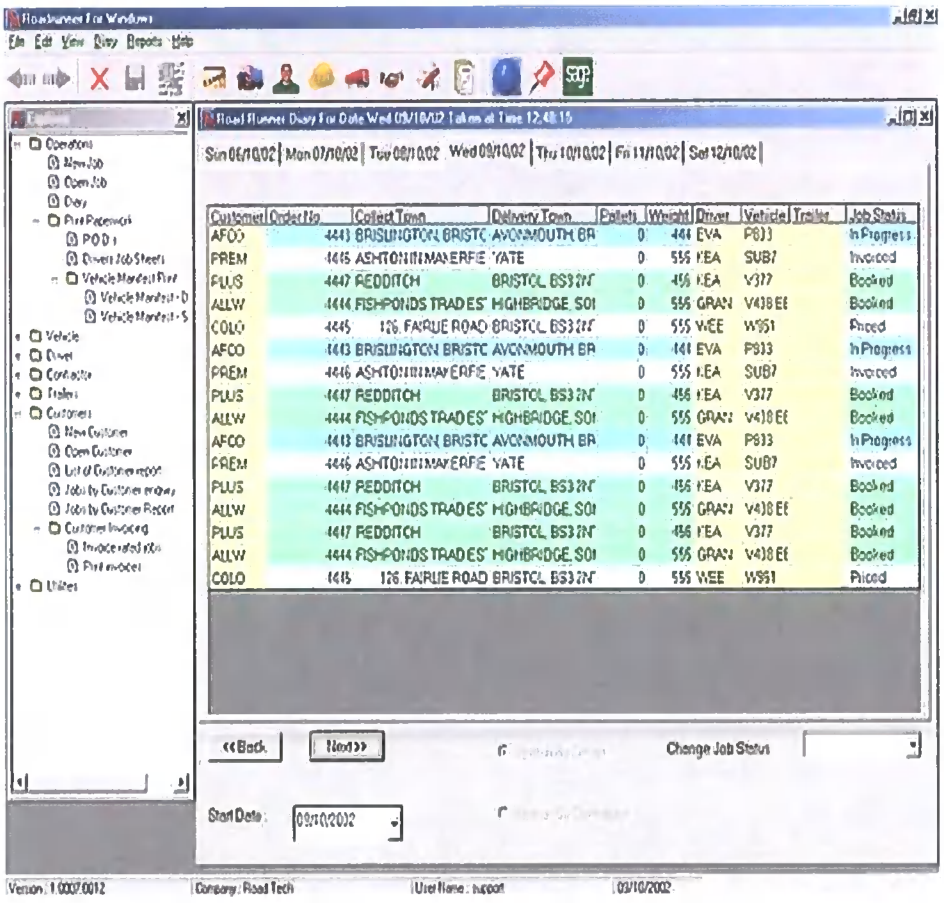
Version: 1.0002.0012 | Company: Road Tech | User Name: support | 09/10/2002

Figure 4.1 R.H.A.'s RoadRunner System. Enter Job's details Screen

Figure 4.1 displays the main screen for the entering of new Jobs. The screen offers many fields for specific data such as order number, driver, booked by and job date, but would have benefited from some more generic fields other than its one comments box. Roadrunner integrates strongly with Sage which as IC were planning on using for their accounting software in the near future, would be beneficial. It also has a number of interesting features such as a virtual pin-board for pre-booking hauliers in advance. This feature was one which had also been requested as part of the functional requirements.

The method of displaying bookings was poor in relation to what IC required. The view used tabbed areas to display the different days, and colours to depict the job statuses. From Figure 4.2 it is possible to see that due to its horizontal linear fashion, it offered no concept of time. For example, if a particular delivery was to take only 3 hours, it would not be highlighted as an opportunity to add a second delivery for that driver in his working day. Ultimately, this would be highlighted when he returned to the depot, and the job completed, but the system did not allow for hypothetical planning.

A good feature was the Explorer style folders to the left hand side of the screen. This removed the need for too many buttons and menu items and enables the rapid exploration of the system. This would make the system simpler to understand for the target user who as previously stated may have difficulty in adjusting to a new system.



Roadrunner For Windows
File Edit View Diary Reports Help
Road Runner Diary For Date Wed 09/10/02 Taken at Time 12:48:16
Sun 06/10/02 | Mon 07/10/02 | Tue 08/10/02 | Wed 09/10/02 | Thu 10/10/02 | Fri 11/10/02 | Sat 12/10/02
<<Back
Next>>
Change Job Status
Start Date: 09/10/2002
Version: 1.0007.0012
Company: Road Tech
User Name: support
03/10/2002

Customer	Order No	Collect Town	Delivery Town	Pallets	Weight	Driver	Vehicle	Trailer	Job Status
AFCO	4443	BRISLINGTON, BRISTC	AVONMOUTH BR	0	444	EVA	P833		In Progress
PREM	4446	ASHTON IIIMAYERFE	YATE	0	555	KEA	SUB7		Invoiced
PLUS	4447	REDDITCH	BRISTOL BS3 2NT	0	456	KEA	V377		Booked
ALLW	4444	FISHPONDS TRADE ES'	HIGHBRIDGE, SOI	0	555	GRAN	V418 EE		Booked
COLO	4445	126, FAIRLIE ROAD	BRISTOL BS3 2NT	0	555	WEE	W951		Priced
AFCO	4443	BRISLINGTON BRISTC	AVONMOUTH BR	0	441	EVA	P833		In Progress
PREM	4446	ASHTON IIIMAYERFE	YATE	0	555	KEA	SUB7		Invoiced
PLUS	4447	REDDITCH	BRISTOL BS3 2NT	0	456	KEA	V377		Booked
ALLW	4444	FISHPONDS TRADE ES'	HIGHBRIDGE, SOI	0	555	GRAN	V418 EE		Booked
AFCO	4443	BRISLINGTON BRISTC	AVONMOUTH BR	0	441	EVA	P833		In Progress
PREM	4446	ASHTON IIIMAYERFE	YATE	0	555	KEA	SUB7		Invoiced
PLUS	4447	REDDITCH	BRISTOL BS3 2NT	0	456	KEA	V377		Booked
ALLW	4444	FISHPONDS TRADE ES'	HIGHBRIDGE, SOI	0	555	GRAN	V418 EE		Booked
PLUS	4447	REDDITCH	BRISTOL BS3 2NT	0	456	KEA	V377		Booked
ALLW	4444	FISHPONDS TRADE ES'	HIGHBRIDGE, SOI	0	555	GRAN	V418 EE		Booked
COLO	4445	126, FAIRLIE ROAD	BRISTOL BS3 2NT	0	555	WEE	W951		Priced

Figure 4.2 RHA's RoadRunner System Diary view

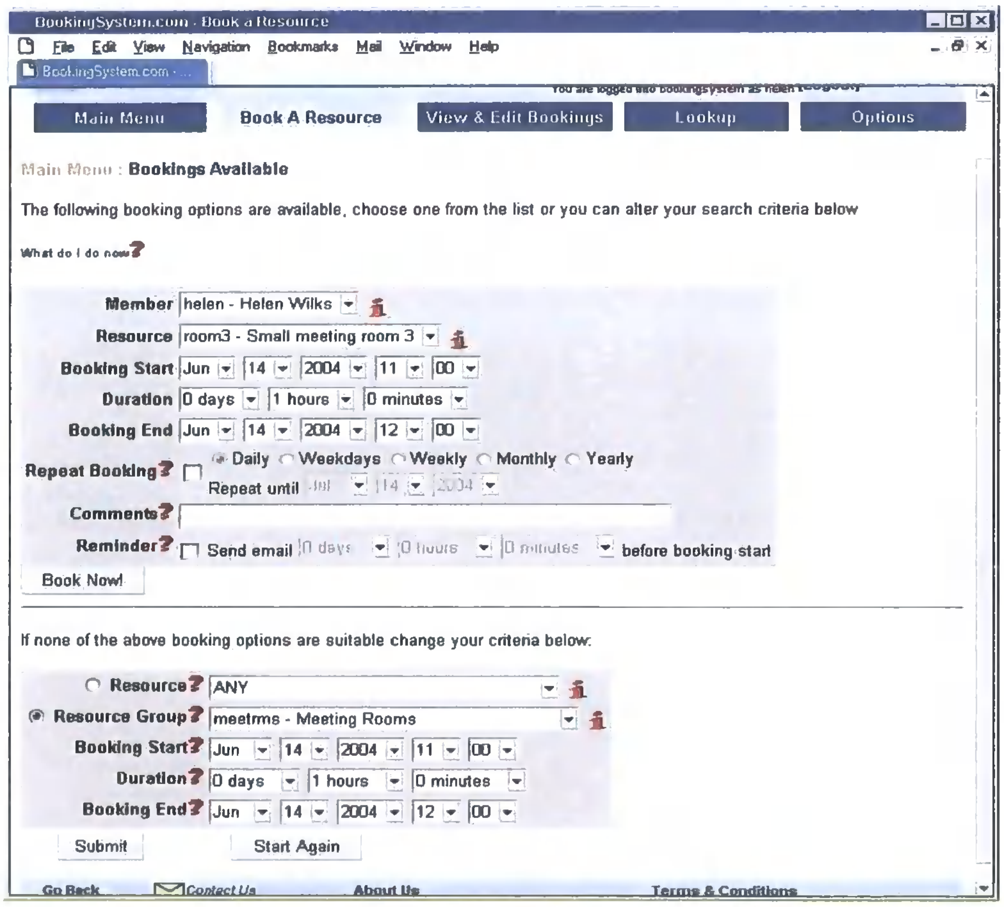
RHA Roadrunner's prospective target users would be those in specialist roles, whereby the users at IC were salespeople who have only reasonable experience in computer use.

A drawback with Roadrunner is that it would not be available if the user was not connected to the company's network. It was worth noting that this would also be the case for any Internet-based system. The offline capability was a desirable function for IC as stated in Requirement 10.1 in Table 2.1. A more distributed offering would be by

RHA Roadrunner's sister product RHA Roadrunner Online which is nearing completion, although this would still not solve the offline problem. The online version was also multi-user.

4.5.2 BookingSystem.com

BookingSystem.com is a flexible Internet based resource management system. It can be configured for access not only by company employees, but also by external associates who may wish to access limited information. As an example, a company who lease business function rooms would be able to allow customers to log in and view available suites. An unlimited set of users can be set up to access the company's data, and several levels of security are available allowing access to different areas of the application.


BookingSystem.com - Book a Resource
File Edit View Navigation Bookmarks Mail Window Help
BookingSystem.com - ...
Main Menu
Book A Resource
View & Edit Bookings
Lookup
Options

Main Menu : **Bookings Available**

The following booking options are available, choose one from the list or you can alter your search criteria below

What do I do now?

Member	helen - Helen Wilks
Resource	room3 - Small meeting room 3
Booking Start	Jun 14 2004 11 00
Duration	0 days 1 hours 0 minutes
Booking End	Jun 14 2004 12 00
Repeat Booking?	☐ [✓ Daily] Weekdays Weekly Monthly Yearly Repeat until Jul 14 2004
Comments?	
Reminder?	☐ Send email 0 days 0 hours 0 minutes before booking start

Book Now!

If none of the above booking options are suitable change your criteria below:

○ Resource?	ANY
● Resource Group?	meetrms - Meeting Rooms
Booking Start?	Jun 14 2004 11 00
Duration?	0 days 1 hours 0 minutes
Booking End?	Jun 14 2004 12 00

Submit   Start Again

Figure 4.3 The Data Entry Screen of BookingSystem.com

With BookingSystem.com, the products that a company is selling are classed as resources. A resource can be anything, in the case of IC, it could be the Caravans. Initially the user needs to set up an account with BookingSystem.com followed by entering a number of preference details. They then need to set a number of options and preference details.

Information for this system can be found at www.BookingSystem.com

Cost

BookingSystem.com is completely free of charge for the basic setup. Charges are applied for alterations. It has no pop-up advertising, no spam emailing, and no selling on of its client details. The owners, Suncrest Systems Ltd made the application viable by offering this light version with a view to promoting extended versions.

Data Availability

Any data added would be stored on BookingSystem.com's. It offered no offline capability. The feature which allowed customer's to access certain resources was interesting. This could be used to allow hauliers to log into the system and check their own timetables.

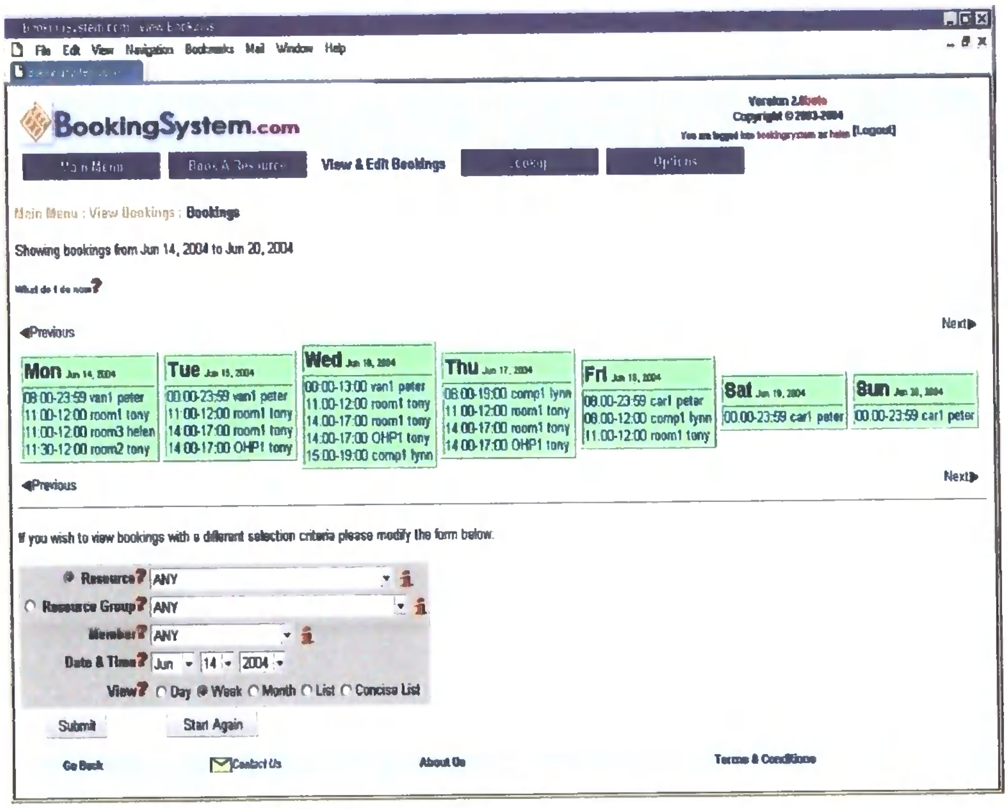
Expandability

BookingSystem.com allows expansion and alteration to their system. Changes to the basic system are made on a per field basis or on a per hour basis for larger jobs. Development can be estimated at £90 per hour, and quotations are available for larger jobs. This would enable the expansion of the system but whether it would be able to expand sufficiently to achieve Ideal Caravan's requirements was doubtful. For example either a large text box, or numerous Yes/No fields were required for information relating to the siting of the van. The Yes/No fields require the development of at least one completely new screen. The level of alterations needed to bring this system into line with what IC required may result in this option not being viable. Further to this, some of the changes needed may be more fundamental to the system and not just cosmetic, and this would discount the option altogether.

Functionality/Interface

The main data entry screen where bookings would be added worked logically and had organised clear displays. Data entry started with entry of the member's details, in IC case, this would be used for the haulier name. Following that, the resources were entered, in this case it would be the van description.

The diary view was clear and it displayed the basic information about bookings ordered by date and time. Each booking could be examined to a greater degree to access further information. This basic view would suit IC who wished to keep the views clear and uncluttered. Unfortunately, the reports available were very limited and only resulted in simple resource usage type reports. To produce the report as required by IC as stated in figure 2.1, for example, illustrating how reliable hauliers were would not be possible without considerable alteration of programming to the system.


File Edit View Navigation Bookmarks Mail Window Help
BookingSystem.com
Version 2.
Copyright © 2003-2004
You are logged into bookingsystem as helen [Logout]
Main Menu
Book A Resource
View & Edit Bookings
Lookup
Options
Main Menu : View Bookings : Bookings
What do I do now?
◀Previous
Next▶

Showing bookings from Jun 14, 2004 to Jun 20, 2004

Mon Jun 14, 2004	Tue Jun 15, 2004	Wed Jun 16, 2004	Thu Jun 17, 2004	Fri Jun 18, 2004	Sat Jun 19, 2004	Sun Jun 20, 2004
08:00-23:59 van1 peter	00:00-23:59 van1 peter	00:00-13:00 van1 peter	08:00-19:00 comp1 lynn	08:00-23:59 car1 peter	00:00-23:59 car1 peter	00:00-23:59 car1 peter
11:00-12:00 room1 tony	11:00-12:00 room1 tony	11:00-12:00 room1 tony	11:00-12:00 room1 tony	08:00-12:00 comp1 lynn		
11:00-12:00 room3 helen	14:00-17:00 room1 tony	14:00-17:00 room1 tony	14:00-17:00 room1 tony	11:00-12:00 room1 tony		
11:30-12:00 room2 tony	14:00-17:00 OHP1 tony	14:00-17:00 OHP1 tony	14:00-17:00 OHP1 tony			
		15:00-19:00 comp1 lynn				

If you wish to view bookings with a different selection criteria please modify the form below.

Resource?	ANY
Resource Group?	ANY
Member?	ANY
Date & Time?	Jun 14 2004
View?	Day [✓ Week] Month List Concise List

Figure 4.4 The Diary screen

Figure 4.4 illustrates the basic but adequate diary view. The system incorporated thorough searching features, although this feature has not been requested by IC.

4.5.3 YourBooking.com

Yourbooking.com was an online booking system. It provided a method by which a user could organise bookings of almost any nature, from villa to video rentals. Further information for this system can be found at www.Yourbooking.com.

Cost

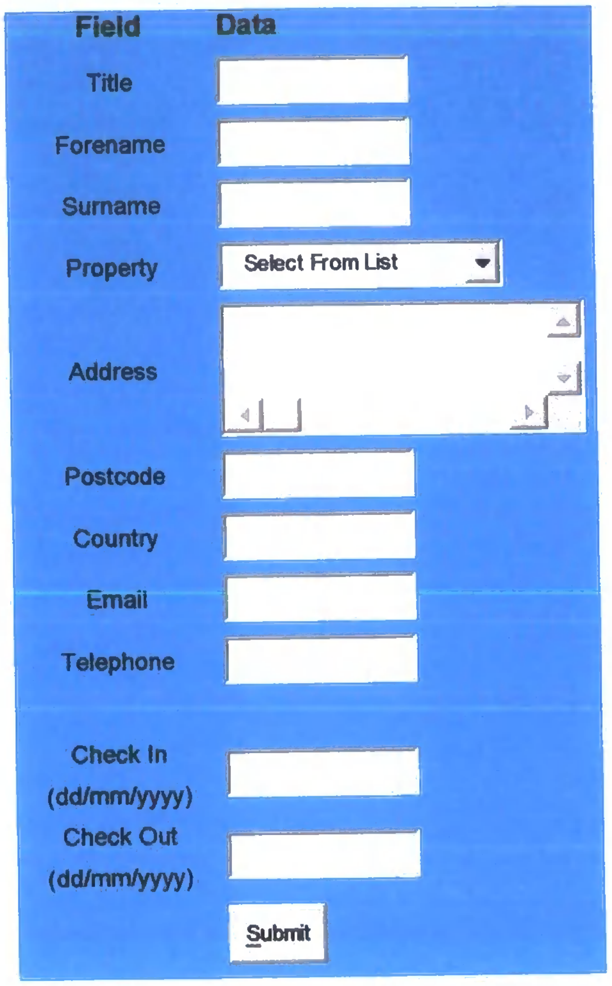
This online system was Open Source [OSC06] (Osc) and so had no costs associated with it.

Data Availability

The data was held on the host's server so there would be no offline capability. Data access was gained via a log-in screen from previously created accounts. Unfortunately, there was no facility to allow differing levels of access.

This system offered a feature where a Customer could access the web site and view available booking options. Unfortunately, this feature offered only minimal feedback, informing the user whether the lodge was available or not. This feature may be capable of handling the availability of the fitters for siting jobs.

If the user wished to make a booking, they had to make a telephone booking. The Customer could not book the lodge online; this seemed to be a missed opportunity.



Field	Data
Title	
Forename	
Surname	
Property	Select From List
Address	
Postcode	
Country	
Email	
Telephone	
Check In (dd/mm/yyyy)	
Check Out (dd/mm/yyyy)	

Figure 4.5 Entering a Customer's Booking in YourBooking.com

The only screen which the general public could access was the enquiry screen. Figure 4.5 shows the actual booking screen which benefits from clarity and simplicity. Here the organisation's staff would enter any bookings taken over the telephone.

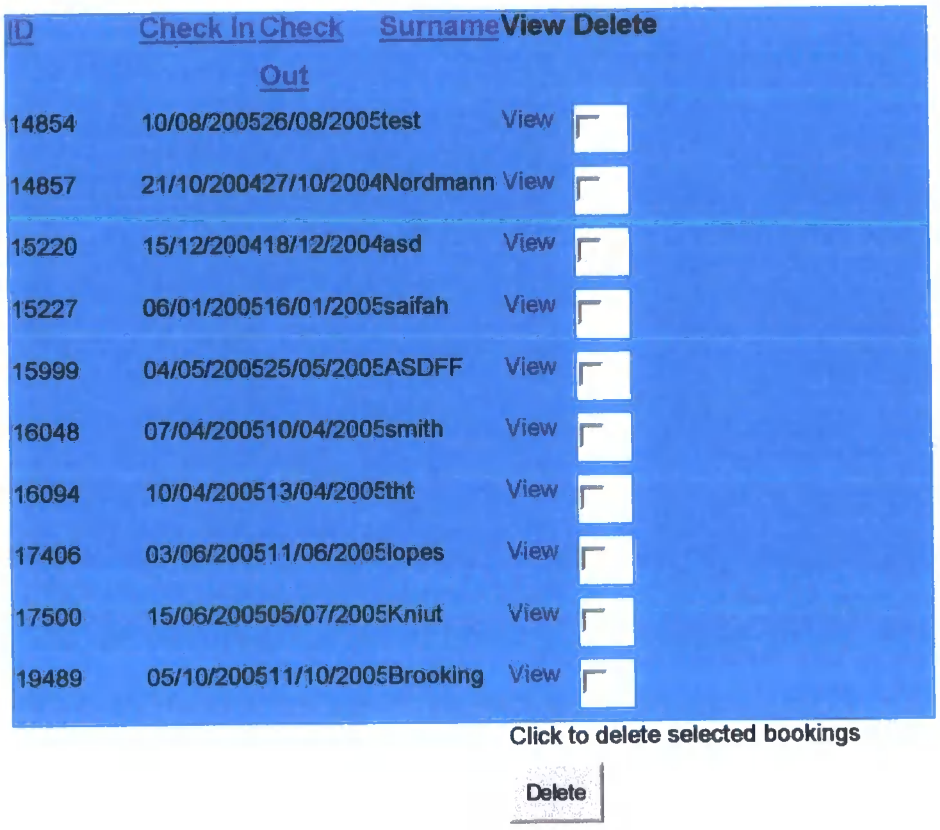
Expandability

A control panel was available to make alterations to the colour scheme, company logos, Internet site links, fonts, field descriptions and field box sizes. No new boxes could be added, nor could the data types be altered. There was no other expandability offered, which seriously affected this system's suitability to IC.

Functionality/Interface

One of the better aspects to this system was the capability of altering the descriptions of any of the fields, along with their sizes and colours. This would offer IC some level of

expansion but not enough to fulfill their needs. There were some ideas not implemented in the previous systems such as the automatic creation of emails to customers to confirm the bookings, currency conversions, discounts, and pre-booking bookings.
The diary view of YourBooking.com was linear and listed all bookings in date order. Figure 4.6 shows its tabular order, which had no time concept. YourBooking.com offered no reporting facilities. The user would have to print out screen shots to get a hard copy of booking details.



ID	Check In	Check Out	Surname	View	Delete
14854	10/08/2005	26/08/2005	test	View	
14857	21/10/2004	27/10/2004	Nordmann	View	
15220	15/12/2004	18/12/2004	asd	View	
15227	06/01/2005	16/01/2005	saifah	View	
15999	04/05/2005	25/05/2005	ASDFF	View	
16048	07/04/2005	10/04/2005	smith	View	
16094	10/04/2005	13/04/2005	tht	View	
17406	03/06/2005	11/06/2005	lopes	View	
17500	15/06/2005	05/07/2005	Kniut	View	
19489	05/10/2005	11/10/2005	Brooking	View	

Click to delete selected bookings

Figure 4.6 YourBooking.com - The Diary view

4.5.4 TrackWhere

Vigo Software Ltd produces a suite of linking applications under the name of TrackWhere. Each application can be purchased as a stand-alone unit or as a combined system. TrackWhere is a comprehensive delivery tracking system which covers all aspects of warehousing and distribution from the design layout of a warehouse, to a suggested packing manifest for deliveries. The application within the suite which was found to be most suitable was called LoadBuilder. This package encompasses the logistics side of warehousing and concentrated on the loading of vehicles, delivery planning and vehicle management. It included numerous features such as vehicle tracking, subcontracting, vehicle maintenance and reporting facilities. TrackWhere can be found at www.trackwhere.com for further information.

Cost

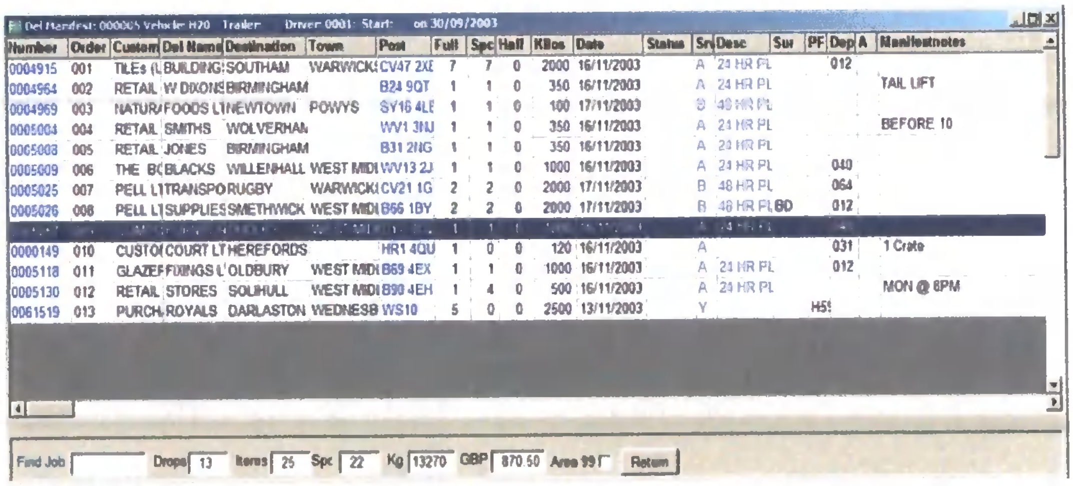
Costs were unavailable for casual enquiries.

Data Availability

The system could be installed onto a single user PC or over a network allowing simultaneous access from numerous computers. As with RHA, Vigo Software were also in the final stages of implementing an Internet based version of the software. It was worth noting that the developers of the two non-Internet based system were both proceeding with the development of an Internet-based system. This might suggest the natural progression of these types of system.

Expandability

After speaking with Vigo Software, it became apparent that there was no possibility of altering any of the fields within the screens to accommodate the unique fields that IC would need. This reduced the software's evaluation score. The new online system however, would introduce the concept of user changes and dynamic screen designs.



Del Manifest: 000005 Vehicle H20 Trailer Driver 0001: Start: on 30/09/2003

Number	Order	Custom	Del Name	Destination	Town	Post	Full	Spc	Half	Kilos	Date	Status	Sr	Desc	Sur	PF	Dep	A	Manifestnotes
0004915	001	TILEs (L	BUILDING	SOUTHAM	WARWICK	CV47 2XE	7	7	0	2000	16/11/2003		A	24 HR PL			012		
0004964	002	RETAIL	W DIXONS	BIRMINGHAM		B24 9QT	1	1	0	350	16/11/2003		A	24 HR PL					TAIL LIFT
0004969	003	NATUR/	FOODS L1	NEWTOWN	POWYS	SY16 4LE	1	1	0	100	17/11/2003		B	48 HR PL					
0005004	004	RETAIL	SMITHS	WOLVERHAM		WV1 3NJ	1	1	0	350	16/11/2003		A	24 HR PL					BEFORE 10
0005008	005	RETAIL	JONES	BIRMINGHAM		B31 2NG	1	1	0	350	16/11/2003		A	24 HR PL					
0005009	006	THE B(	BLACKS	WILLENHALL	WEST MIDI	WV13 2J	1	1	0	1000	16/11/2003		A	24 HR PL			040		
0005025	007	PELL L1	TRANSPO	RUGBY	WARWICK	CV21 1G	2	2	0	2000	17/11/2003		B	48 HR PL			064		
0005026	008	PELL L1	SUPPLIES	SMETHWICK	WEST MIDI	B66 1BY	2	2	0	2000	17/11/2003		B	48 HR PL	BD		012		
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	1	1	[illegible]	[illegible]	[illegible]		A	24 HR PL			[illegible]		
0000149	010	CUSTO	COURT LT	HEREFORDS		HR1 4QU	1	0	0	120	16/11/2003		A				031		1 Crate
0005118	011	GLAZEF	FIXINGS L	OLDBURY	WEST MIDI	B69 4EX	1	1	0	1000	16/11/2003		A	24 HR PL			012		
0005130	012	RETAIL	STORES	SOLIHULL	WEST MIDI	B90 4EH	1	4	0	500	16/11/2003		A	24 HR PL					MON @ 6PM
0061519	013	PURCH	ROYALS	DARLASTON	WEDNESB	WS10	5	0	0	2500	13/11/2003		Y			H5!			

Find Job: Drops: 13 Items: 25 Spc: 22 Kg: 13270 GBP: 870.50 Area 99

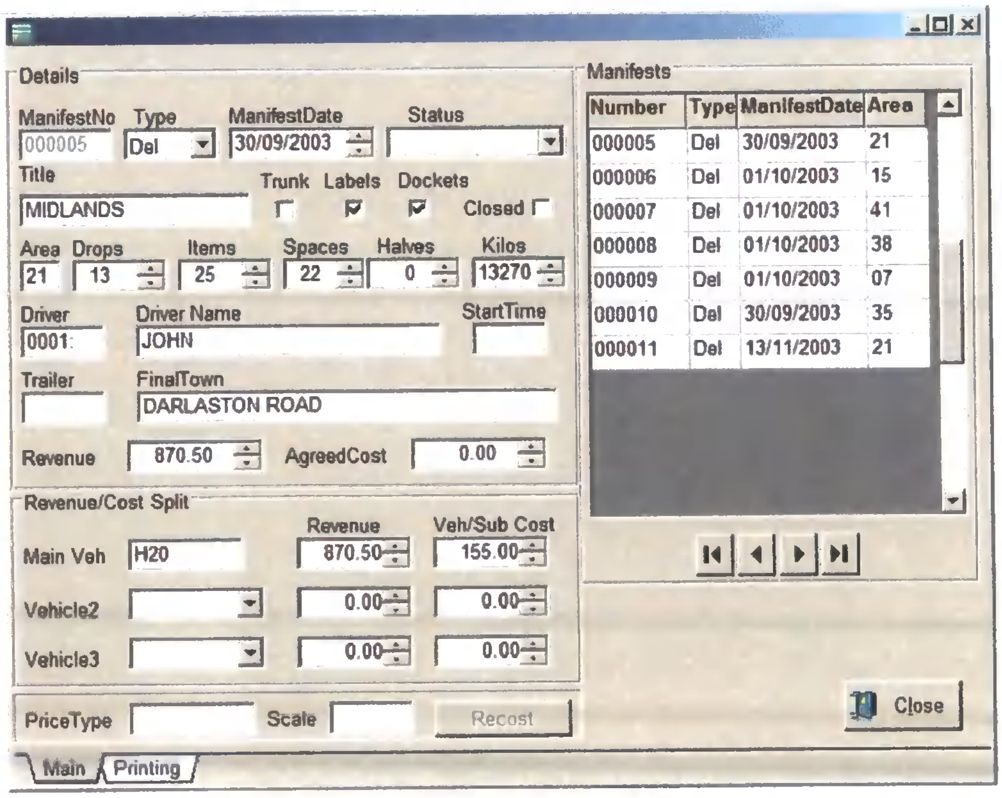
Figure 4.7 The LoadBuilder Bookings Overview Screen of LoadBuilder.

Functionality/Interface

Figure 4.8 shows that the diary view was linear, similar to that found on RHA's Roadrunner. One of the better features of this design was the ability to pre-book jobs, and to differentiate between pre-booked jobs and actual jobs as they were displayed in a different colour. Figure 4.7 also illustrates the fact that there was no concept of time within the booking screen which would have been beneficial to IC.

The data entry screen, Figure 4.8, was found to be confusing. A more logically laid out screen might have started with the date for example, or the driver name. A closer examination of the layout helped in understanding its design, but it may have been more suitable for the information on the screen to have been placed over two screens instead of one.

TrackWhere had good reporting facilities, allowing a number of editable templates to be used. It also had a clear menu negotiating system and other facilities which were not needed by IC such as the ability to incorporate satellite tracking, the creation of emails which would automatically inform the customer as to the status of their job, and optimised loading suggestions. The email facility would have been useful to IC in notifying their Customers as to the status of the van movement.



Details

ManifestNo: 000005 | Type: Del | ManifestDate: 30/09/2003 | Status:

Title: MIDLANDS | Trunk ☐ | Labels ☑ | Dockets ☑ | Closed ☐

Area: 21 | Drops: 13 | Items: 25 | Spaces: 22 | Halves: 0 | Kilos: 13270

Driver: 0001 | Driver Name: JOHN | StartTime:

Trailer: | FinalTown: DARLASTON ROAD

Revenue: 870.50 | AgreedCost: 0.00

Revenue/Cost Split

		Revenue	Veh/Sub Cost
Main Veh	H20	870.50	155.00
Vehicle2		0.00	0.00
Vehicle3		0.00	0.00

PriceType: | Scale: | Recost

Manifests

Number	Type	ManifestDate	Area
000005	Del	30/09/2003	21
000006	Del	01/10/2003	15
000007	Del	01/10/2003	41
000008	Del	01/10/2003	38
000009	Del	01/10/2003	07
000010	Del	30/09/2003	35
000011	Del	13/11/2003	21

Figure 4.8 The LoadBuilder Data entry screen.

Figure 4.8 was found to be illogical and could have been split into sub-screens; for example, the financial information could have been removed from this screen and placed on its own screen.

Further Options

There were also a number of software companies specialising in haulage organisational software on a much larger scale. This is a niche market and the prices reflected this fact. Although their costs would mean that they were not a viable option, they would be worth at least viewing to see what features they could offer.
Fleet Computer Services Limited was one such company and has been supplying hardware and software to transport companies for over 20 years. They offered a product called Traffic Office system which can be found at http://www.fleet-computers.co.uk/Page_1x.html.

4.6 Comparison and Outcome of System Reviews

A comparison of the systems was made difficult because of the large variations in their costs, and that the systems were aimed at differing target markets. Referring back to the criteria which the review was based upon in Table 4.1 assists in this process as it differentiates aspects of the systems. Table 4.2 shows a summary of each system against the set criteria.

	RHA Roadrunner	BookingSystem-.com	MyBooking.com	TrackWhere
Criteria 1 Cost	Reasonable. £100 per user	Reasonable. £100 initial fee for a one user license. £100 per user thereafter	Free	Unknown
Criteria 2 Data availa-bility	Needs to be distributed over a network. No off-line availability. Internet version due out shortly	Needs to be distributed over a network. No off-line availability	Needs to be accessed via the Internet. No off-line availability.	Accessed over a network. No off-line availability. Single or multi user.
Criteria 3 Expan-dibility	None	Mainly cosmetic	Cosmetically good, functionally poor	None. Future Internet-based system may be more suitable.
Criteria 4 Key Func. Req.	Achieved some of the essential functionality and some of the desired. Good data entry but poor summarisation of bookings	Achieved some of the essential functionality and some of the desired. Good data entry. Limited reporting facilities	Achieved some of the essential functionality and some of the desired. Clear data entry but poor summarisation of bookings	Achieved much of the essential functionality and much of the desired functionality. Data entry confusing

Table 4.2 How each System measured up Against the Crieria

It is possible to review the above data by examining each criterion in turn:

1 **Cost**. MyBooking and BookingSystem were both free which sets them apart from the chargeable systems. The mid range system was Roadrunner which would cost £1000 for 10 users. Finally, Trackwhere was reviewed which would arguable be the most costly of the systems.

2 **Data Availability**. Data availability was provided through 2 options; over the Internet or over a network. MyBooking, BookingSystem were Internet based whereas Roadrunner and Trackwhere were both network based. This reflected the costs comparison.

3 **Expandability**. Both Internet versions offered only cosmetic or minor alterations. Trackwhere and RoadRunner offered no realist expandability.

4 **Functionality/Interface**. Trackwhere and Roadrunner's functionality were nearer to what was required, than the Internet versions, but they were still not sufficient without alterations. The two Internet systems had better data entry and diary views.

As can be seen from Table 4.2, none of the systems were clearly suitable. The main problems encountered for the reviewed systems were:

- None offered both online and offline capability.
- A lack of expansion; free fields which could be renamed, or the ability to add new fields.
- Poor reporting facilities.
- Poor data entry and data viewing screens.

The most suitable system was TrackWhere, although this was still considered to be insufficiently suitable as it had not met all of the required criteria for functional requirements as listed in Table 2.1. The following Table 4.3 illustrates how TrackWhere compared against those requirements.

Req. No.	Applies to	Functionality	R	D	Req. met?
1.1	Hauliers details	Allow add, edit and delete	Y		Y
1.2.1	Bookings	Allow add, edit and delete	Y		Y
1.2.2		Check for double bookings	Y		Y
1.2.3		Record information relating to the siting of the caravan once it has been delivered.	Y		N
1.2.4		Record information relating to the transport arrangements.		Y	Y
2.1	Users	Password protection	Y		Y
2.1.1		Password should be checked against IC main database	Y		N
2.2	Paper-trail	A log to record changes to the system data		Y	Y
3.1	Diary screen	This should look like a diary. It should have daily and weekly views.	Y		Y
4.1	Diary screen	Allow users to pre-book fitters and hauliers		Y	Y
5.1	Data entry form	Needed clarity and simplicity.	Y		Y
5.1.1		Direct the user though key steps of the booking.	Y		Partial
6.1	General	Should be designed with the company's Corporate Image in mind	Y		N
7.1	Access to data	Must be accessible at all company offices.	Y		Y
8.1	Existing data	Use company's database for park names	Y		N
8.1.1		Use company's database for passwords	Y		N
8.2.1	Application	Must be compatible with existing network	Y		Partial
9.1	Hauliers	Showing reliability and accident rate		Y	Y
9.2	Users	Showing credibility of bookings, number of incomplete jobs etc		Y	Y
10.1	Entire System	The data should be available either offline or online.		Y	N

Table 4.3 Comparison of requirements against TrackWhere

As a result of TrackWhere not being sufficiently suitable to IC, a compromise was considered; would it be possible to create an add-on to one of the reviewed systems to augment it to the level of what IC needed? After consideration, it was felt that the time it would take to create an add-in system and achieve only partial success of requirements would be ill-proportioned to the time taken to create a completely new system and achieve full functionality. Having reached this conclusion, the second option needed to be considered; this was to create and implement a new system. Furthermore, the experiences gained from reviewing the existing system can be used to direct the design of any new systems.

5. The Experiment

5.1 The Question of the System

In the previous Chapter, the viability of purchasing a piece of COTS for IC to solve a set of problems within their CPs was examined. After looking at a number of options, none of which proved to be suitable, it was decided that a new system would need to be developed. This new system would need to take into account current CPs, the current computer systems, staff abilities and the requirements of the new system.

From a technical point of view, there were two main system options:

1 A Microsoft Windows based system using the company's Intranet, similar in design to those systems currently in place.

2 An Internet based system running on the company's Intranet.

There were numerous advantages and disadvantages of the differing technologies. These are listed in the following table:

Discussion	Technology	Advantages	Disadvantages	Comment
Functionality	Windows	Ease of programming. Good development tools. Familiar G.U.I.		The company were experienced in Windows development
	Internet	Quicker querying	Not as easy to program as Windows/ GUIs not as expected	Lack of experience in designing/deploying on Internet platform
Network	Windows	Self managing – network is internal – Secure	Physical restrictions Larger network footprint	Can be upgraded if necessary
	Internet	Unlimited network geography Many new technological opportunities such as external access, PDA access	Availability dependant upon connection	
Development	Windows	Known technology	No new skill learned	
	Internet	Opportunity to try new technology	Research needed – will impact on time	

Table 5.1 The Advantages and Disadvantages of differing Technology

Table 5.1 illustrates that both technologies offered benefits through their use, but came with associated overheads. For example, the Internet version would offer better expandability and portability, but with a greater risk due to the lack of knowledge the company had in this field. Glass [Glass98] stated that of all project runaways, 45% of them were a direct result of the technology being new to the organisation. This was a concern to the company management and would need to be included in the risk assessment.

A further point to note was that each technology would offer differing standards and methods both during their creation, and in their finalised product. Certainly the two systems would offer the same core functionally, but each may offer secondary functionality which would be unique to each technology. An example of this might be the use of 'Tool Tip' text, or keyboard short cuts in a windows application which may not be available in a web page. This needed to be considered.

Referring back to the aims of this thesis which sets about examining the identified problematic business process, and investigating possible solutions to the problem, it might be beneficial to take advantage of these highlighted technological benefits and incorporate as many as possible into the new system. The thesis will also consider the methods by which the company identifies, appraises and implements new computer systems and the actual creation of a new system would be an ideal starting block for such a study.

After further discussion, it was suggested by IC that the two systems could be developed in parallel and at some point in development, the more suitable be chosen. This did not seem a viable proposition as it would almost double the development time and therefore might compound or at least extend the time that the company would continue experiencing the problems within the domain. It was also suggested that the company use this opportunity to instantiate a set of guidelines for use in the development of future in-house systems. They would take this opportunity to create a more structured method of software development which could be applied to all future projects. This could be used in future as the company were expecting to expand and were looking to implement at least two more larger-scale systems in the future. The dual development approach could be used to satisfy the structured method definition.

5.2 Risk Assessment

As with a proposal of any new software system, there were a number of risks involved which needed to be assessed before commencement. Risks needed to be acknowledged and documented if they were to be controlled, and the process of analysing these risks, once established could be applied to all future developments.

A method of risk analysis needed to be introduced. McGraw [McGraw04] defined Risk Analysis as "a way of gathering the requisite data to make a good judgment based on knowledge about vulnerabilities, threats, impacts and probabilities". Risk analysis is not only about gathering the data; it is about being able to act upon it which in itself is a risk.

Previously, the company management had no concept of formal risk assessment with regard to software development. They understood the principles of risk assessment when applied to other areas such as health and safety where assessment is tangible. It is more difficult to pre-empt a risk which is not necessarily physical. Whichever risk assessment was used, it would need to be assumed that the first attempt would be less than perfect.

Many researchers have tried to address the problem with risk assessment; Boehm [Boehm88] stated that "Relying on risk-assessment expertise, the spiral model places a great deal of reliance on the ability of software developers to identify and manage sources of project risk".

There are numerous methods such as Kemerer's Function Points Measurements [Kemerer96] which introduce the notion of using metrics and checklists in order to formalise risk assessment. To varying degrees, most formal software risk assessment methods unfortunately do not take into account the ability and experience of the assessor as part of the evaluation process. Nor do they take into account the number of cycles of the process needed to achieve the proffered solution.

As this would be the first time for IC to make a formal assessment of this kind, the process would be a learning experience as well as guide for future projects.

Most of the risks found in Table 5.1 could be addressed and minimised purely by designing the system around them. To this end, identifying them reduced the relevance of these risks. Other areas such as project over- run and functional over-requirement cannot be eliminated but can be managed. Were the project much larger, it would have been possible to apply Nogueira's model of risk assessment for the Evolution of Requirements [Nogueira98], which uses metrics to measure the expansion and contraction of a project of a period of time. However for a smaller project such as this, it would be sufficient to utilise Boehm's Risk Management Plan [Boehm89] where he proposes that a top 10 list of risk items should be identified. If these were decided upon and together with estimated impacts, they could be assessed at a later date as to how they affected the duration and quality of the project. Those risks identified are listed in Table 4.1.

Risk	**Detail**
Incorrect system design	Design does not solve problem domain issues
	Design creates other problems
	System fails to gain Management backing and is therefore dropped
Failure of Technology	The system does not work adequately through utilising untried technology
Project over-run	Lack of experience at managing project
	Functional over-requirements
	Requirements creep
Security risks	New systems allows intrusion
	Loss or alteration of data
System too technical	The system is too difficult/complex for target user to understand.

Table 5.2 **Areas of Possible Risk**

5.3 The Collection of Metrics

During the process of designing and implementing the new systems, metrics would be collected. These would be used at the review stage to compare the development of the two systems. It is worth being reminded that the success of the system did not just lie in its ability to fulfill the functional requirements, but also in the analysis of themethod by which it was created. The reason for the collection of measurements throughout this process was summarised by Briand et al [Briand96] in that if offers "an effective means to understand, monitor, control, predict and improve development and maintenance projects." It also provides the opportunity for the process to be recursively improved through documentation. One of the problem aspects to this is the skills required in analysing and documenting such measures successfully. Not all attributes are readily quantified; measuring the quality of a piece of software is an example of such.

A metric can be defined as an entity which can have an attribute assigned to it. One of the better abridged definitions is that of Texas State Libraries [TSL06]; "A random variable x representing a quantitative measure accumulated over a period".
Fenton and Neil [Fenton00] argue that where organisations introduce the collection of metrics into their software development process, they almost always are motivated by one of two activities:

1 The desire to assess or predict effort/cost of the development process.
2 The desire to assess or predict the quality of software products

They also classified metrics into 3 entities:

- Processes: Any software related activity such as testing
- Products: Any artifact or deliverable from a process such as specifications and code.
- Resources: Items that are used within a process such as personnel and software tools.

Each entity may have internal and external attributes associated with it.

The metrics and the methods by which these metrics are collected needed to be identified. The discovery of the measurement activity can be achieved in a number of ways. Goal/Question/Metrics (GQM) is one such method that was proposed by Basili [Basili85] in 1977. He refers to it as an informal method and would be suitable for smaller projects. GQM operates on the principle that if a goal is identified, then questions can be asked relating to that goal. This ultimately leads to a metric which can be used to assess the risk.

The project goals are multipartite in as much as it is not only assessing the success of an implemented system, but also the success of the comparison of two similar systems, and also the successful creation of a structured business method and CP.

Table 5.2 lists the criteria which metrics shall be based. These are the factors which could affect the design and development process. These factors were chosen for their suitability from the combined lists of what Kemerer [Kemerer96] and Boehm [Boehm88], and were combined with others such as iteration times which were felt suitable to the project but missing from the combined work. Further, there are entries on their lists which have been excluded from Table 5.2, such as entity relationship complexity and functional complexity which when placed into the context of the project were considered not to be of significant benefit. There were a number of models, which may have been used to consider theses criteria; for example Harrison and Cook's [Harrison87] Micro/Macro method, or by examining the system architecture as in Henry and Kafura's [Henry81] Information Flow model. Other metrics such as stability could only be gathered at a later date.

No.	Metric	Measurement	Internet Based System	Windows System
1	Software/ Hardware research	Time		
2	Software/ Hardware installation	Time		
3	Database Creation and implementation	Time		
4	Lines of code	Lines of code		
5	Initial coding time	Time		
6	Testing	Time		
7	Coding due to testing/faults	Time		
8	Additional requirements	Quantity		
9	Coding due to additional requirements	Time		
10	Iteration Times	Time		
11	Total time taken	Time		
12	User's system grading			

Table 5.3 The Metrics Collection Table

Each point in Table 5.2 is explained below:

1 **Software/Hardware Research**. This would be the time taken to identify any new software or hardware needed to allow that particular system to be implemented. An

example of this would be the choice of Apache, the Internet server software needed so that the UNIX server could supply Internet pages at request.

2 **Software/Hardware Installation**. This would be the time taken to install any new hardware or software required for the new systems. An example of this is the installation of the mySQL database application on the UNIX server.

3 **Database Table Creation and Implementation.** The time taken to analyse and create the databases for the systems.

4 **Lines of code.** The number of lines of code needed to complete each system. This is one of the most basic metrics used in measuring the complexities of a project. By comparing this with other elements of the systems such as number of entities, functions and process flows, Zhao [Zhao98] highlights that it is possible to create a metric relating to the complexity of the system.

5 **Initial Coding Time.** The time taken to write the code throughout the iterations in order to complete the first release of the systems.

6 **Testing.** The time taken within the iterations to test the written code.

7 **Coding due to Testing/faults.** This would be the time taken to write any additional code or make amendments to existing code as a result of the testing stage.

8 **Additional Requirements.** This will be the number of additional requirements added during the development process after the design had been agreed.

9 **Coding due to Additional Requirements.** The time taken to implement any new code relating to additional requirements.

10 **Iteration Times.** This will be the breakdown of times spent on each iteration.

11 **Total Time Taken.** The total time taken to produce the completed system.

12 **User's System Grading.** The overall score, graded from 1 to 5, given to the system by IC from reviews.

To be able to use the results of this experiment as a benchmark for future developments, these metrics needed to be collected and presented at the conclusion of the experiment in a valid manner.

5.4 Development Methodology

IC had identified two functional requirements which they classed as important in the new system; the provision of a simple user interface and the simplified displaying of data. Certainly there were other important points to the system, such as data synchronisation, but IC was making these suggestions based on what they knew and understood of the system. With this in mind, a suitable development lifecycle was sought. What was needed was an iterative process, with a fairly quick start-up where the user interface could be seen to be being developed correctly whilst the underlying more important aspects be implemented as well.

IC previous systems had been developed off site, and there had been no form of prototyping offered. When a version of the system was finally brought in to be demonstrated to the company, it was fairly advanced and no option was given for backtracking on fundamental design.

Boehm [Boehm88] proposed that his Spiral Model for software development lifecycles worked on an iterative basis allowing the prototyping of a system. As the iterations took place, the system could be developed to a deeper and deeper level until the functionality was completed. This method was suitable as the basic system could initially be created and iteratively be improved. This prototyping method would enable a basic design of a system to be created allowing the two key requirements to be successfully implemented. This could then be reviewed by the company and would give the designer the opportunity of implementing factors which the company did not necessarily identify initially as important. Wasserman [Wasserman80] proposed this method of system design whereby lower level features of the system could be deferred until later in the project. This is a good method from the system designer's point of view due to its flexibility. It was noted that this could become a problem issue if IC realised that the planning of certain areas of the development had not been finalised, and so wished to force a number of design changes through at a later date. This was a change management (CM) issue and would need to be catered for.

Figure 5.1 illustrates Boehm's Spiral model.

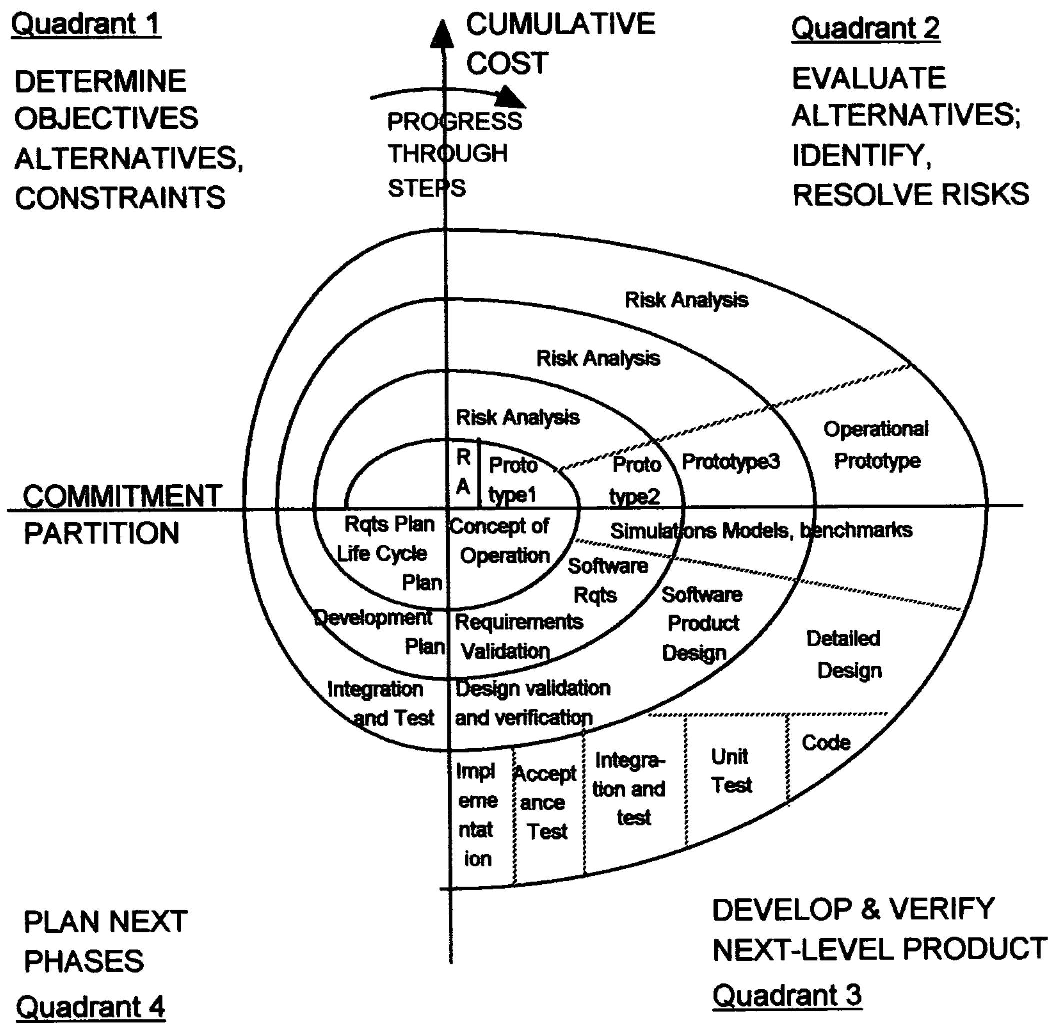


Figure 5.1 Boehm's Spiral Model for Software

Figure 5.1 represents how the Spiral Model allows the iterative assessment of risks and metrics (top right hand segment). Other lifecycles such as The Waterfall method are less well defined in these areas. As the collection of metrics was so important in the experiment, changing the company's procedures to adopt the spiral model is a significant step forward.

5.5 GUI Design Considerations

The design of a Graphical User Interface (GUI) is deemed by most users as the most important factor in the system development. Most users are generally not involved in the defining of system functionality, reporting, security etc, and the only input they may have into the system may be in the design of the GUI. The consequence of this is that failure to implement a GUI to the requirements of the user's community may mean they perceive that the whole system is 'incorrect'. Changes to GUI standards may have a consequence for the company's training requirements. Inexperienced users might need to be trained in the use of the new GUI to be convinced that just because it is different does not mean that it is unacceptable. Goransson [Goransson86] suggested that more research should be conducted into this phenomenon identifying that most computer users were not computer users but tool users and that users first introduced to new GUIs need careful management.

IC had stated that they felt it important that the GUI should be clear and easy to understand. This requirement had been included in their KFRs in point 5.1 of Table 2.1 With reference to Meyer [Meyer03], who states within his design principles that a successful system should involve:

- Small interfaces. GUI design should attempt to keep the amount of information on display at any one time to a manageable level.
- Explicit interfaces. A GUI should display information relevant only to that process being modeled.
- Information hiding. The GUI should not show unnecessary information. An example of this is at the point a Hotelier places a booking onto a system, the system should concentrate only on the booking, and not display the Customer's details.

The above points had all been requested by IC in respect to their GUIs.

5.6 Summary

It was decided to create two different versions of the system utilising differing technologies. Each system would be developed using Boehm's Spiral Model, and would provide an opportunity to examine this method of system development and document a more structured method of software engineering for IC future developments.
The risks associated with the systems needed to be collected and assessed. Metrics would also be collected and reviewed in an attempt to understand, monitor and control the process of software engineering.

6 The Bespoke Systems

Typically a software system can be developed in a number of ways. One of the challenges for IC and for any SME is the selection process for each development option that arises. In IC the development options are typically concerned with the selection of hardware platforms or software. If costly mistakes are to be avoided then it is important that the development team select an option that is robust and appropriate for the situation in which the system will be deployed. Knowledge gained from the decision making process is useful in informing the future decision making processes and so through the application of the development process, the company will collect decision data for the first time and make this available for future projects. Unfortunately however, it is difficult to truly evaluate decisions and therefore inform future development practices as in the majority of cases only one of the development options has been carried forward. Therefore ordinarily no data is available for comparison to be made.

6.1 System Design Options

With the decision that a new system was required came a set of questions associated with their development, such as:

1 Which OS should the development be based on?
2 Which programming language should be used?
3 How was the system to be deployed?
4 How was the data to be stored and synchronised?
5 What is a suitable GUI?

The following section will consider each point in turn. Where the options are more complex, QOC is used to illustrate the decisions made. In other options, even though the QOC approach has been applied, due to the simplicity of the decision making process, it may not have been represented.

6.1.1 Operating Systems

The company was using MS Windows exclusively for their PCs and servers except for one older UNIX server which had been installed and maintained by a third party. All company employees were comfortable using the MS Windows environment.
Despite this fact, the company where open to suggestions of other technologies, especially those which utilised the Internet. They had considered the future of IT within the company and were eager to take this into account. The question was, which OSs were most suitable to support the systems.

The two main OSs which were considered were Microsoft Windows and UNIX/Linux both based on PCs. A third technology which the company had expressed an interest in was the use of PDTs such as Blackberrys, but as the company was not at that stage yet, the suggestion was kept in mind for future developments. UNIX is distributed under a number of OSc; Linux, GNU, Sco UNIX, Uniq and Fedora Linux.

One system would be developed on a Windows platform and the second on a UNIX/Linux platform. It was decided that due to its user friendly and high level of functionality, RedHat Linux would be used. It came supplied with Apache, a piece of software which enables the server to serve Internet-pages at request. This collection of software would enable the UNIX server to operate as a complete web server.

6.1.2 Programming Languages

Commercial programming languages have been around since the early 1960s [Rosen67]. Over this time, they have developed from simple calculators to highly complex graphically displayed development environments. Theoretically, when a system is to be developed, the features of the system are to be taken into account and the programming language chosen on this information.
Typically, a language would be chosen because of a variety of issues such as;

1 Costs associated with the programming language: A company may already have purchased a particular programming language compiler for previous projects. Rather than have the additional cost of purchasing a new compiler, they may opt to use the one they already have.

2 Skills/Experience/Historical: If a company has previously developed software in a particular language, they may be more willing to use this language again due to the lower risks involved rather than switch to a new one which would require new skills to be learned.

3 Code library: Companies build up libraries of commonly used code. An example of this would be a user environment interface; a company may reuse the same user environment for a number of projects [Doublait97].

With SME's, often practical considerations take priority when faced with these decisions. Each OS offered a number of programming language options. A decision needed to be made upon which programming language was most suitable for each of the two OSs. The following two Sections provide a list of those options.

6.1.2.1 UNIX Programming Languages

To make an informed decision, as to which UNIX programming language was most suitable, the most commonly available options needed to be considered. The most suitable options were short-listed, and these options were reviewed against a set of criteria as listed in Table 6.1.

	Question: Which UNIX/Linux programming language was most suitable?
Option 1	Active Server Pages
Option 2	Cold Fusion
Option 3	CGI with C or C++
Option 4	DHTML
Option 5	Java Server Pages
Option 6	PHP
Option 7	Perl
Option 8	Tool Command Language
Criteria 1	Cost: The relevance of cost should be considered.
Criteria 2	Database support; How well the language supported and was supported by databases.
Criteria 3	Support: How well the language was supported in general by then Internet, user groups and publications.
Criteria 4	Functionality: The level of functionality that the language offered. For example, HTML alone would not provide sufficient capability to support the required functionality as listed in Figure 2.1.
Criteria 5	Speed. Although speed was not critical, certain languages may operate substantially faster than others.

Table 6.1 The Options and Criteria relating to the suitability of the Programming Languages.

	Active Server Pages	Cold Fusion	CGI with C or C++	DHTML	JSP	PHP	Perl	TCL
Cost	5	5	9	9	9	9	9	2
Database Integration	7	4	6	3	6	8	7	4
Support	3	3	8	6	5	8	8	3
Functional -ity	8	7	7	6	6	7	7	7
Speed	4	7	7	6	5	6	6	7

Table 6.2 A Table representing a review of the suitable of the Programming Languages

Table 6.2 illustrates that the most suitable languages were PHP and CGI. If CGI were chosen, it would need to be used in conjunction with a language such as C or C++. Further to this, if PHP or CGI were used, the system could be distributed using Internet technology. IC could use their current Intranet to host their own Internet based system. This would be a huge advantage when it came to the synchronising of data. It was decided that Option 3, CGI would be used as it offered a solid programming language with good database and Internet-page support.

6.1.2.2 Windows Programming Languages

The majority of the company's development experience involved the use of Microsoft's Visual Basic. There are not many Software Development Kits (SDKs) which offer as complete a solution to programming as VB. This is a development tool which allows the programmer to create programs which look as if they are part of the Windows OS environment. This would help maintain uniformity throughout the company's system and also meant that there would be very little in new skills needing to be learned. The current version of VB being used was Version 6 Visual Studio. One of the key reasons for its popularity is the ability to produce professional looking systems very quickly. It also offered facilities for databases and report writing.

The Windows version of the system would be developed using VB for these reasons.

6.1.3 Databases

Whichever database was chosen, it would be hosted on either one of two servers. If it was a Windows OS database, it would be held on the IBM 2000 server, and if it was a UNIX OS database, it would be held on the UNIX server. Both servers were located at the Durham office.

The question of which database was most suitable for the two systems needed to be considered. As reviewing databases can be a time consuming process, and due to time constraints and the simplicity of the data structure, it was agreed that the database options should be limited to the best known and documented ones.

6.1.3.1 UNIX Databases

There are a number of databases suitable for use with PHP and CGI under UNIX/Linux. These databases could be split into two main types, those available as OSc and those which were sold as commercial ventures. According to The Linux Journal [Lin1] and Linux Magazine [Lin2], the most commonly used databases for Linux were PostgreSQL, MySQL and Oracle. These three databases were consistently in the top 5 Linux databases in their reader's surveys and so were considered in the QOC table 6.3 below.

	Question: Which UNIX/Linux database was most suitable?
Option 1	PostgreSQL
Option 2	MySQL
Option 3	Oracle
Criteria 1	Cost: The relevance of cost should be considered.
Criteria 2	Compatibility with CGI and PHP: The database should be supported under the programming platforms already chosen.
Criteria 3	Support availability: Support such as books, web-sites and tutorials
Criteria 4	Ease of use: Installation, set up and administration of the database

Table 6.3 A Table Illustrating the Database Options and Criteria

There were other criteria which could have been included such as query speeds, and the level of complexity in indexing, but it was felt that for what was needed, these criteria would be sufficient.

	PostgreSQL	MySQL	Cost
Cost	Open Source: Free	Open Source: Free	Free as Beta package
Compatibility	Yes	Yes	Yes
Support	Yes	Yes	Limited
Ease of use	Installation wizard and fairly easy database creation	Installation wizard and easy database creation	Installation wizard, but more complex database creation

Table 6.4 Table Illustrating the Suitability of the Databases

PHP and CGI/C++ both benefited with being well supported by user communities, in print and on the Internet for PostgreSQL and MySQL. They were both very similar in respect to installation, set-up and for querying.

Option 2, MySQL was chosen based on recommendations from business associates with experience of using it.

6.1.3.2 Windows Databases

There are numerous databases available to Windows, too many to consider which would be suitable for this application.

Therefore, it was decided not to review the Windows databases and that MS Access would be used. This was for the following reasons:

1. MS Access had been used for all previous bespoke systems for the company.
2. The company had licenses to use MS Access without further cost.
3. The simplicity of the data structures needed for the system would easily be catered for with MS Access.

6.1.4 Methods of Synchronisation

The method of synchronisation would be the key to the whole project. It would be unsuitable having a booking system with out-of-date data. The question was which method of data synchronisation was most suitable? Table 6.5 illustrates this.

	Question: Which method of data synchronisation was most suitable?
Option 1	Packet transfer: This method relies on the master database keeping a record of all changes made to it, and sending those changes out in packets to a set of distributed systems periodically. This would entail each distributed computer holding a copy of the data locally. There must be some mechanism for synchronising data in both directions.
Option 2	Direct database access over an internal network: Using direct links to the database, applications could display a live version of the data.
Option 3	Querying MS Access using TCP/IP: The distributed application could make a query request on an ad-hoc basis for data. This method would not allow offline access to data, but benefited from the data being available from any location with Internet access.
Option 4	An amalgamation of options 1 and 2: A local copy of the data would be held on the distributed machines. When online, the local database would be synchronised periodically with the master database, by way of flags informing the different Users that changes had occurred.
Criteria 1	Data made available at all offices
Criteria 2	Data made available online
Criteria 3	Data made available offline
Criteria 4	Data made available both on and offline

Table 6.5 Table showing the Options for Data Synchronisation

Table 6.5 lists the options available for the synchronisation of the data, which should be made available at any time to authorised users. It should also be made accessible from anywhere in the physical confines of the company. The requirements in Figure 2.9 stated that the data must be accessible to all company offices.

	Packet transfer	Direct database	TCP/IP	Amalgamation
All	Yes	No	Yes	Yes
Online	No	Yes	Yes	Yes
Offline	Yes	No	No	Yes
On & offline	Yes	No	No	Yes

Table 6.6 A Table showing the Suitability of the Options

Figure 6.6 displays the suitability of the options when compared against the criteria. The option for the Internet system was Option 3 which utilised SQL queries against the main database over TCP/IP. This would keep the database programming in PHP to a minimum and the built in security which comes with TCP/IP could be utilised.
The most suitable option for the Windows version was Option 4, which although would incur more programming, it would allow both on-line and off-line capability.

6.1.5 Designing the Graphical User Interface

A system's GUI is an important aspect in the success of a system and needs to be considered fully. Table 2.1 listed the KFRs and it showed that IC had also placed importance on the GUI. Requirement 6.1 stated that the GUI should reflect the company's corporate image. It should also be clearly laid out and uncluttered.
Both systems should utilise the lists of parks, hauliers and users in the existing IC database. The diary view should represent as close a replication as possible of a normal diary including the navigation aspects.

Drop down lists, calendars and quick text functions should be used where possible to minimise data entry. Screens should address the requirements as stated in the W3.org website and that this issue required further investigation.

The W3.org Internet site [Berners-Lee06] which is an international consortium for Internet standards has this to say about design standards:

1 Content must be **perceivable.**
2 Interface elements in the content must be **operable**.
3 Content and controls must be **understandable**.
4 Content must be **robust** enough to work with current and future Internet technologies.

Figure 6.1 W3.org's Web site Design Standards

It was decided that for the Internet system, hypertext screens would be sufficient to fulfill the company's requirements. The company decided that the factor of time and cost be of prime importance, but noted that the selected option would not preclude the development of a GUI at a later date.

Therefore in summary, the design of the system would adhere to these points at minimum, and where possible attempt to replicate the views and layouts of IC existing systems.

6.2 Other Factors

There were other factors which would affect the system development, and they needed to be considered. Such factors might include:

1. Set up costs. There may be unforeseen costs such as additional software licenses, additional hardware and upgrades.
2. Set up time. The time it would take to set up a Web server was unknown and this needed to be considered.
3. Time taken to learn the new technology/language. Again, this was unknown prior to implementation.
4. The future of such technologies. I.e. is it worth going to the extent of learning a new technology such as Tcl for a system development? Would Tcl be of any use in the developer's future?
5. The future of the developer. Which technologies would the company's developer benefit from learning?
6. Available support. Which technologies are strongly supported, well documented and the availability of contacts experienced in those technologies.

Figure 6.2 Other Factors affecting the System Development

6.3 Identifying Changes to the Company Roles and Procedures

The introduction of the new system and managed business process had a number of high level benefits to the company management.
These included:

1 The greater compliance to company procedures. The company would be gaining a system which would guide the user through a set of steps which would have to be taken before a transport booking could be made. This would be a substitute of a formal CP. In effect, the system would be ensuring that all bookings were made whilst adhering to the CP.

The introduction of the system would mean that the current methods of operation might need to be altered. Kettinger [Kettinger97] pointed out that "BPR is increasingly being considered as a transformation of the organisational sub-system". This is in comparison to Hammer's [Hammer90] obliteration concept. What was being attempted was to alter a sub-domain problem within the company with a view to affecting the whole domain.

2 Ensuring a minimum set of booking details were collected prior to commitment. This was achieved by the system rule that no bookings could be placed without a core of essential information being provided to the system.

3 More even parity of treatment for full and part-time employees. This was a result of the staff hierarchy being leveled with regard to the booking of transport. Previously, the company had noted that certain members of the sales team, notably those employed on a part time basis or newer members were having their transport arrangements shuffled to suit others. This was clearly not appropriate. If a caravan is sold, and a member of staff has arranged with the customer to have the transporting and siting of that caravan on a pre-arranged day, then it should not be possible for this to be altered except in extenuating circumstances. At that point, only that member of staff or the administrator should be allowed to alter the booking.

4 The removal of certain roles. With the implementation of the new system would come the removal of the transport representatives' roles. This role as described in Chapter 2.3 would be welcomed by the transport representatives, as they would no longer be needed to do work on behalf of another member of sales staff. It also meant that they would only be responsible for the accuracy of their own transport arrangements, not for the whole department.

The company would benefit from the opportunity to produce reports. They would be able to create reports which highlighted trends such as which sales members were more organised than others, and the failure rate of van movements.

5 More efficient team working. There may be intangible benefits such as the sales team working smarter. They would need to have more information at hand, and therefore be more organised at the point of making a booking. This may set a precedence of also improving standards elsewhere in the business.

6.4 Resulting Overview

Each sub-section of the two systems were to be created in parallel. For example, both sets of database tables were to be created concurrently. This would allow a comparison to be made more easily and reduce bias of the collected metrics, for instance, consideration had to be taken of the fact that whichever version's function was attempted first. This system may have a slight overhead in time taken to complete it due to the learning process.

System	Physical deployment	System set-up	Development
Internet	To be installed on an existing UNIX server which was connected to the internet by a firewall server	RedHat Linux would be installed to provide an O.S. which the rest of the installation and set-up could be managed through. The server would use apache to serve web-pages. Further software such as PHP and MySQL was needed.	The scripts for the web-pages would be developed with a Windows version of PHP for ease of accessibility, and then transferred for the web-server for testing. As the web-pages may be available to the public, each web-page must be self authorising.
Windows	The Windows system was to be distributed over the existing company network. It would utilise the main Ideal database for park detail s and login rights. The completed system would be installed on to each individual PC on the network and would use IP addressing which would enable any user to access the database data.	The only form of setup would be the installation of the application software onto each PC and the creation of the relevant tables on the master database.	The code would be developed within the Windows VB environment and then be transferred and installed on to 2 other P.C.'s for testing purposes. After testing, the system would be deployed company wide.

Table 6.7 The System Implementations

Table 6.7 describes the method by which the two systems would be implemented.

6.5 Summary

This Chapter discussed the options available for the development of the two new systems. It considered the OS's, and the programming languages. It identified the most suitable databases for the chosen platforms, and considered the best methods of keeping the data synchronised. It also considered the impact of GUI which was an important aspect of the development. After the most suitable options had been chosen, Section 5.4 gave an overview of the two proposed systems highlighting any key points.

The chapter also identified what changes might take place within the company's operating procedures once the chosen system had been implemented.

7 Evaluating the two new Systems

The purpose of evaluating a project where a piece of software has been created is to not only evaluate the software but also the way it was created and implemented. Kumar [Kumar90] states that "the primary reason for such evaluations seems to be project closure and not project improvement". He suggests that in most cases within business that the personnel involved in the analysis of a project are those who have the most to lose if the project is not deemed a success. The result being that the evaluation is a means to which the project can be assessed as to its completion. Kumar [Kumar90] goes on to say that successful evaluation of a SE project should be more concerned with looking for improvements in the project as a whole. By this method projects can be iteratively improved.

The evaluation process shall be broken down into 3 steps:

1. To evaluate and compare the two new systems. In this step the two systems were introduced to the company. After an evaluation period, they were reviewed by staff and management and preferences collected.
2. To evaluate the methods used in the analysis, design and implementation of the new systems.
3. To assess the use of existing and modified development methods (as in step 2) to create a structured business method of software engineering combined with a reusable framework which IC can apply to any future software developments.

7.1 An Evaluation and Comparison of the two New Systems

There are a number of formal methods and tools which can be used in order to evaluate software against the needs of an organisation. One such method was proposed by Ubhayakar [Ubhayakar03]. Ubhayakar offered a framework of evaluations from which to base a formal comparison of requirements against the entities and objects of a completed system. Although based upon military systems, his framework centers upon a specification language called the Prototype Verification System (PVF). By creating a script of PVF based upon a system's requirements, specification and design, a

verification process will specify the correctness of those inputs. This is an expansive process which would be used on systems such as those which are safety critical.
A second method was proposed by Kontio [Konti95] et al which introduces a formal screening process based on the organisations' documentation. During the evaluation for IC use, it was found that despite Kontio's claim to it having a small footprint; it was felt to be too complex for a smaller organisation to benefit and was aimed at large software developments. The questions/ options/ criteria QOC [McKerlie93] method would also provide a suitable approach for an SME to select from a set of alternatives. However, the difficulty with the use of this method resides in the selection of appropriate evaluation criteria upon which the decision making is to be based. For this reason, the QOC approach would be adopted and two feasible development options taken through implantation. This meant that the decision making process could be evaluated and a more accurate assessment made of appropriate evaluation criteria to apply to decision making in the future. When considering any system development options using the QOC method (as highlighted previously), the difficult aspect is ensuring the appropriate questions and criteria are used. Some researchers have published examples that can be applied to this work such as McKerlie and MacLean [McKerlie93], but even they make a point of stating the difficulty faced with QOC is in the experience required in identifying the appropriate questions. A significant contribution of this work will be an evaluation of the decision making process through the implementation of two development options so that the predictions and final outcome can be compared. The interesting feature of all this research work is how little they focus on NFR, instead focusing on functional aspects of decision-making systems. A further contribution of this work will more closely follow the impact of NFRs on the decision making process.

The two systems were implemented on a parallel basis for trials between the periods of the 7th of November 2005 to 30th of December 2005. The decision on which of the two systems was to be used by the company was to be scheduled for January 2006, and the chosen system to be implemented by February 2006.

To make an evaluation of the two systems, the following points must be considered:

1. A comparison against key functional requirements.
2. A comparison against non-functional requirements.
3. The system users' opinions.
4. The chosen system

7.1.1 Comparison against Key Functional Requirements

The KFR's were as stated in Table 2.1. Both systems were designed specifically using this set of guidelines. Both systems achieved complete success in achieving the functionality required in the Required column.

Requirement 7.1 of table 2.1 stated that data should be available from all company offices. This was not achieved; there was a difference in the way the data could be accessed at the Waren office. The Waren office was not connected to the company's external network; there was only an ISDN line present. The result of this was at this location:

- The Windows system could only offer an offline version of the data.
- The Internet version could offer a slower but live version of the data.

This meant that the Windows version did not quite succeed in achieving all of the functionality in the Desired column in as much as the data might be out of date, but due to the use on this site, this can be judged as a minor failure. Therefore, choice between the systems would come down to fulfilling the NFRs and personal choice, and hence allow detailed evaluation of the ability of the QOC approach to represent and accurately portray for decision making the selection of development options through consideration of NFRs.

7.1.2 Comparison against Non-Functional Requirements

In comparison with the functional requirements, the non-functional requirements were more abstract and covered areas such as the system costs, expandability and reliability.

Table 4.1 showed the non-functional requirements. Point 4 of 4.1 was decomposed previously into the functional requirements as stated in Table 2.1 previously. It listed the non-functional criteria required for a successful system which are reviewed as follows.

7.1.2.1 Cost

As expressed, the cost of the system was not paramount, yet was relevant. The company was expecting value for money; the cost should be relative to the usefulness of the system. It is important to consider how each system would compare when examined from a cost point of view.

In a bespoke system cost may be broken down into time; analysis, design, coding, implementation and maintenance amongst other things. Briand [Briand02] writes that the main factor in costs associated with software development is the size of a project. This may seem obvious, but there are other contributing factors such as the complexity of the analysis and mode of system deployment. There are other aspects such as cost of tools, hardware and software, but in this case, these were less important due to their availability as considered in Chapter 4.4.

Neither system needed any new expensive software or hardware purchasing to be implemented; therefore, the major cost issue would be dependent upon other aspects such as development and maintenance time. Although the cost of maintenance is not known yet, by viewing the metrics collected, it was possible to see that the Internet based version took the longest time to create and implement. This made it the costliest option from the point of view of time taken. The Internet based version cost approximately 47% more than the Windows version based on time taken; see Figure 8.6.

As was previously stated, the importance of cost is relative to the effectiveness and usefulness of the system. Therefore if the Internet based system was found to be a considerably better system, the longer development and implementation time could be considered as worthwhile. The answer therefore lay in the decision of the preferred system which shall be considered further into this chapter.

7.1.2.2 Data availability

The requirement for data availability had first been introduced in Chapter 2 where point 4 of Figure 2.8 stated that the system must "Ensure that that information was made available to authorised Employees when it was needed".

The Windows version allowed bookings to be added whether online or offline. The data would then be synchronised when the user next went online. The system was able to highlight where the synchronisation process would cause a double booking. Ideally, the booking which was created first of the two should be honored, but in this case, the existing booking on the master database was honored. This meant that off-line booking were of a lower priority to those placed on the master database; this could be conceived by the following rules:

1 Online bookings were confirmed bookings.
2 Offline bookings were provisional bookings until data synchronisation.

This offline feature offered a distinct advantage over the Internet version allowing employees to continue to work away from the offices. This requirement emerged as almost redundant as the logs showed that after the initial trials, the offline version was never used; it seemed that users preferred to operate with live data. Whilst the Windows version was online, it operated in a similar way to the Internet version.

The only other difference between the two was that the Internet version offered live data away from the company's external network, for example a salesperson would be able to access it from home if they had an Internet connection.

The criteria in the Table 2.1 stated that the data should be accessible from any of the company's locations. Both systems offered this functionality but in differing ways. The Window's system offered a slightly better data service by allowing the user to create bookings offline and synchronise the data later.

7.1.2.3 Expandability

The main aspect of expandability is to ensure that the system is sufficiently adaptable so that it can be easily enhanced to support any future changes in direction that the company may wish to make.

The exact concept of expansion could be achieved in a number of ways such as spare fields, buttons and using tables rather than fixed options could be considered as offering expandability. Further to this, the expansion of the system might depend upon available knowledge to be able to expand the system using the language it was developed in. The Window's version was created in VB which is a widespread language, and the company should have no problems finding a capable programmer who could continue with any system expansion.

The Internet based system, which had been developed using PHP may prove to be more difficult in its expansion. PHP is more of a specialty development language, and is more limited in its scope in certain areas such as GUI design than other environments such as VB.

On the other hand, as a proposal was being considered whereby management and staff who worked out on site were to be issued with PDA in the near future, the company could foresee new developments within the business that would disperse its operations to further sites. With this in mind, the Internet based version which could be used from any location with Internet access would be a distinct advantage to the company.

The outcome of this was that the Internet based version offered the better options for expansion, whereas the Windows version offered the better tools through which to expand.

7.1.2.4 Key Functionality

This was discussed as part of 7.1.1.

7.1.3 The System Users' Opinions

This section has been separated into two areas:

- The first is the inclusion of a questionnaire completed by users establishing a points system to quantify an appraisal of the two systems.
- The second is a discussion from differing perspectives as to the successes and failures of the two systems.

7.1.3.1 The Questionnaires

The staff at IC were requested to run the three systems in parallel: the original paper system, and the two new computerised ones. The 2 computerised systems would create log files during their utilisation, listing user's usage periods. This would be useful in recording the times taken to complete comparable work. At the end of the trial period, each member of staff and management which had taken part in the trial was asked to complete a questionnaire. A summary of the results from these can be found in Figure 7.1.

Research highlighted that the questionnaire could be presented in many different styles, but given the use of QOC and the user community, it was decided that in order to gain a precise outcome, the questionnaire would be based on metrics. Foddy [Foddy94] stated that "questionnaires are expressions of attitudes, feelings, and opinions rather than factual accounts of past behavior and interactions". He went on to say "The relationship between what respondents say they do, and what they actually do is not always strong". This is understandable in as much as it is part of human nature. But Foddy also states that "[with questionnaires] there is a lack of clear conceptualism of what is being measured". In other words, the questioned person does not fully comprehend the concept that he/she is being questioned over. This seems to reflect a lack of understanding by the body setting the questions as opposed to the persons being questioned. This is backed up by McColl [McColl01], who wrote about this; "individual survey researchers need to take into account the aims of the particular study, the population under investigation and the resources available".

It is less subjective to request measurements using this form of assessment, as opposed to 'views' which can mask an accurate outcome and be hard to accurately analyse. Heath [Heath06] states that "it is possible to design survey instruments in such a manner that one can probe specifically for certain benefits and improvements", and goes on to say that "gathering clearly defined metrics" (within a questionnaire) "benefits more accurate results". With that in mind, it was noted that the questionnaire will only be as accurate as the questions are relevant and clearly understood by IC workforce.

The responses to the questions offered were to be in the form of a scale from 1 to 5, 5 being the most positive response. The system users were given a set of 7 questions, and the system supervisors were given a set of 4 questions. All of these involved rating various aspects of the system.

The questions given to the staff/system users were as follows:

How easy was it to learn the system within the given period of time?
This question was asked in order to establish how easily the users felt they had learnt the new system. This would be useful in gauging the success of the GUI design, and in comparing the two systems, it may be possible to ascertain those points which worked and those points which didn't.

How easily could you negotiate around the system?
This question had similar implications to the above in as much as the responses could be used in judging how effective the differing methods of negotiating the system were deemed.

How would you rate the data entry screen?
As the effectiveness of the data entry screen was one of the key points of the system, it would be useful in knowing if the screen was considered to be productive.

How would you rate the diary screen?

As the diary screen was such an important aspect to the system, it was felt necessary to include a metric by which it could be judged against.

How would you rate the speed of the system?

It was not imperative that the system be very fast, but it should offer at least acceptable responses. When users are working, they do not like waiting for screens to be populated and the response to this question would illustrate the user's expectation of this.

How would you rate the overall functionality of the system?

This question was asked to establish if the system offered sufficient functionality. Establishing what functionality was missing, if any, would be discussed later.

What would be your overall score for the system?

This figure would enable the system to be judged overall, and would be useful in establishing the big picture when all of the responses were collated.

The questions given to the management/system supervisors were as follows:

How well do you feel the system has been received by staff?

It is important that an organisation knows that its staff can work comfortably with an application; there would be a tendency for users to shy away from applications which they did not like working with. By including this question, it would be possible to establish how readily the users could accept the system.

How would you rate the overall functionality of the system compared with your expectations?

This question is important in establishing the level to which the company felt the system was functionally complete. If the system had been purchased, this question may have related to value for money.

How reliable do you feel the system will be?

This is an important question for the company's managers. They could only make a judgment in relation to the time the system had been in operation.

What would be your overall score for the system?

As with the question to the users, this figure would enable the system to be judged overall, and would be useful in establishing the big picture when all of the responses were collated.

The questionnaires can be found in Appendix B.

The following table, Table 7.1 is a collation of the responses to the questionnaires.

Staff Questions	**Internet system**	**Windows system**
How easy was it to learn the system within the given period of time?	17	16
How easily could you negotiate around the system?	18	16
How would you rate the data entry screen?	19	14
How would you rate the diary screen?	16	13
How would you rate the speed of the system?	16	14
How would you rate the overall functionality of the system?	18	15
What would be your overall score for the system?	19	18
Sub Total	**123/210**	**106/210**
Management Question		
How well do you feel the system has been received by staff?	8	7
How would you rate the overall functionality of the system compared with your expectations?	9	7
How reliable do you feel the system will be?	7	6
What would be your overall score for the system?	9	7
Sub Total	**33/60**	**27/60**
Total Score	**156/270**	**133/270**

Table 7.1 The Questionnaire Results

There were 6 staff given the questionnaire, therefore the maximum score which each question could receive collectively would be 30 points (5 points x 6 people). There were 3 people given the management questionnaire, therefore the maximum score which each question could receive collectively would be 15 points. The results in Table 6.1 showed that the Internet based version of the system gained the highest percentage of points available.

System	Staff responses	Management responses
Internet system	59%	50%
Windows system	55%	45%

Table 7.2 The Scores Achieved as a Percentage of Maximum

It is also worth noting from Table 7.2, the figure of 59%. This was the highest either of the two groups would offer the better of the two systems. Had the company been approached initially with the offer of the development of a system which their own staff would rate at only 59%, it is unlikely that the project would ever have proceeded. This point could be related to chapter 1.1 where previous attempts at bespoke systems had resulted in the company 'never [having] been satisfied with the results'. IC were asked this question and although the concept is quite a complex one, they felt that that a successful system in their view would need to achieve at least 80% of its core requirements.

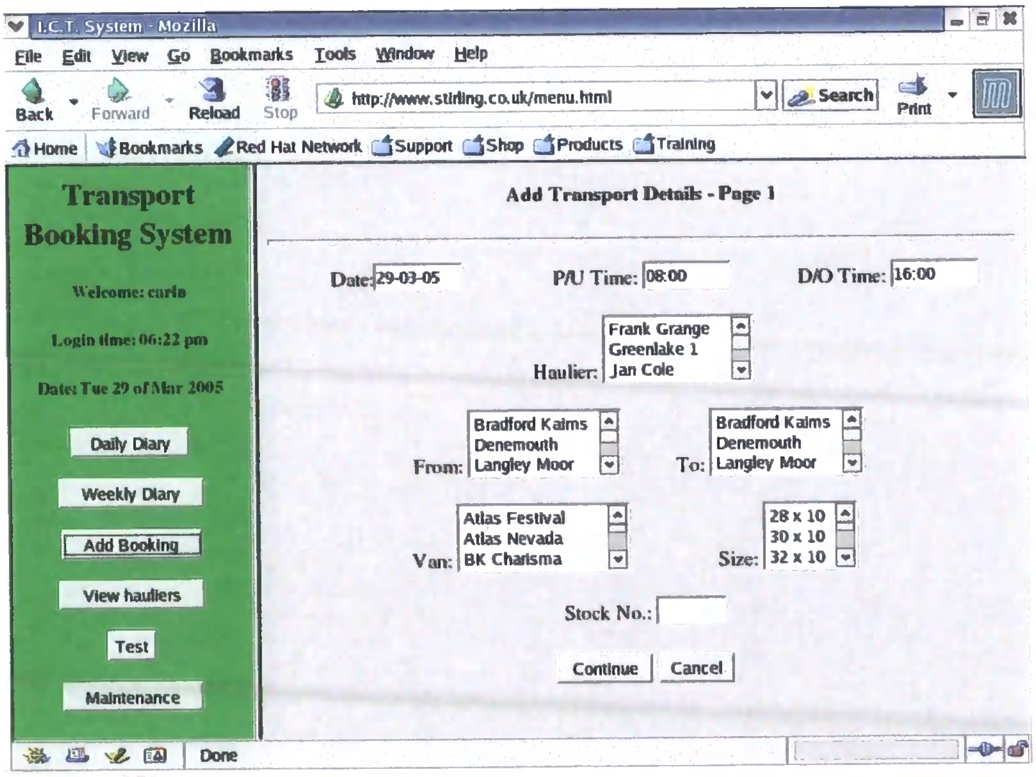
7.1.3.2 Differing Perspectives

Metrics are useful to quickly quantify an outcome, but may not show the user's real thoughts behind the system. To gain this understanding, a cross-section of the users were requested to give their own thoughts on the two systems. The users selected were those with more experience in using PCs. This was mainly done with the hope that their perspectives would hopefully reflect in their experiences of I.T. These users were requested to use both systems in parallel using live bookings as data and to keep a diary of their observations. When a transport arrangement needed to be placed on the system, it was entered onto both systems at that time. They also checked the systems on a daily basis observing how the other users were utilising the system. At the end of a two week period, they were each asked to summarise their observations with any relevant recommendations. The following two sections evaluate the two versions from the differing perspectives.

7.1.3.2.1 The Internet Based Version

The following section highlights the outcome of the evaluation of the Internet based system comparing the differing perspectives of staff and management.

Staff . A selection of the target users, in this case, the company's staff were asked to provide observations of the two systems. They were asked to consider amongst others; GUI design and form layout, reliability, ease of use and overall perception.
The staff provided mainly positive feedback about the system. They liked the form layouts due to their simplicity. They did state that they would have preferred a menu bar rather than a series of buttons for negotiating screens. The diary screen was generally accepted as satisfactory, if basic.


I.C.T. System - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://www.stirling.co.uk/menu.html
Search
Print
Home Bookmarks Red Hat Network Support Shop Products Training
Daily Diary
Weekly Diary
Add Booking
View hauliers
Test
Maintenance
Continue
Cancel
Done

Transport Booking System

Welcome: carla

Login time: 06:22 pm

Date: Tue 29 of Mar 2005

Add Transport Details - Page 1

Date: 29-03-05 P/U Time: 08:00 D/O Time: 16:00

Haulier: Frank Grange, Greenlake 1, Jan Cole

From: Bradford Kaims, Denemouth, Langley Moor

To: Bradford Kaims, Denemouth, Langley Moor

Van: Atlas Festival, Atlas Nevada, BK Charisma

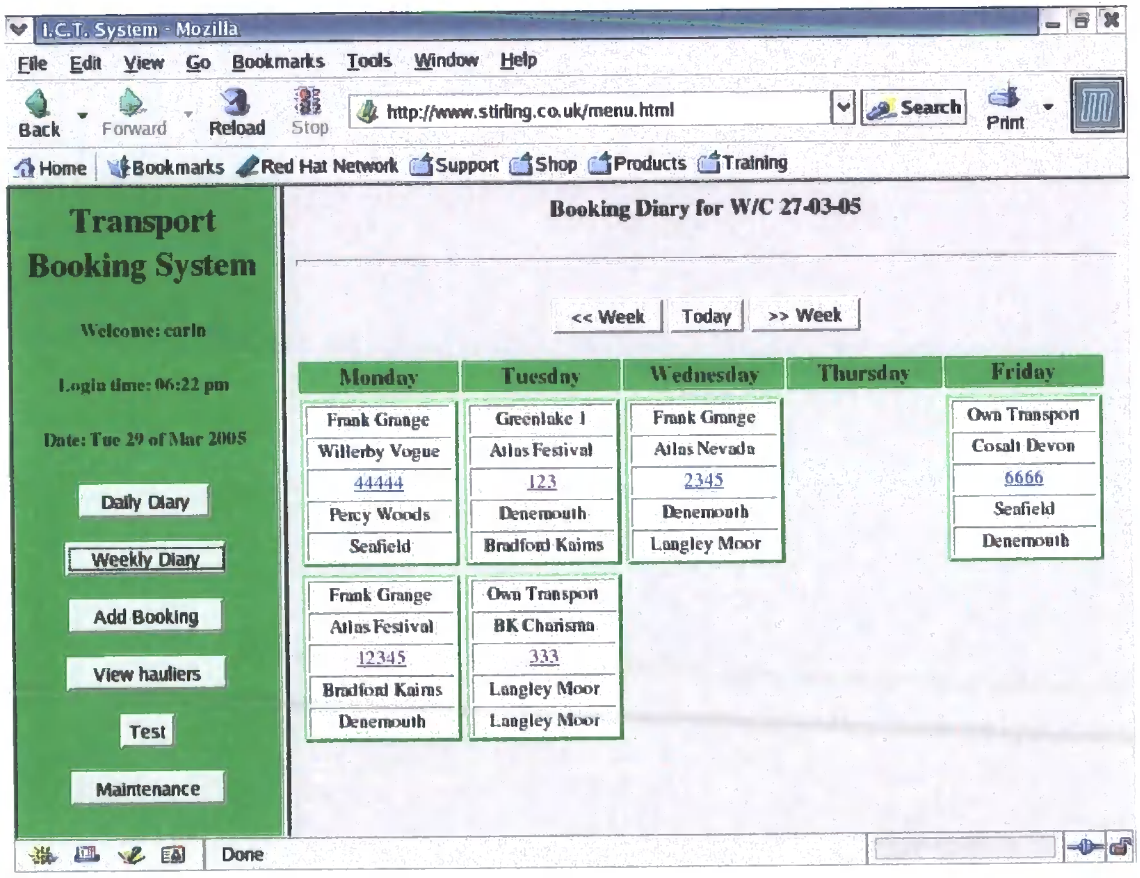
Size: 28 x 10, 30 x 10, 32 x 10

Stock No.:

Figure 7.1 Page 1 of the Data Entry Screen

The system was evaluated at Morpeth, Durham and a number of the other locations. Because of the method by which scripting works, data entry had to be broken down into a set of pages, one of which can be seen in Figure 7.1. The result of this was that the amount of information the user needed to add to a single page did not exceed 10 pieces of data, unless it was purely a yes/no tick box. This made the pages look more manageable for the staff and were simpler to learn.

A menu bar could not be offered on this version of the system due to the use of HTML for displaying the page. One alternative would have been the use of a drop-down box listing the available menu options. The diary view which had highlighted as being only satisfactory was also as a result of this.



Booking Diary for W/C 27-03-05

<< Week | Today | >> Week

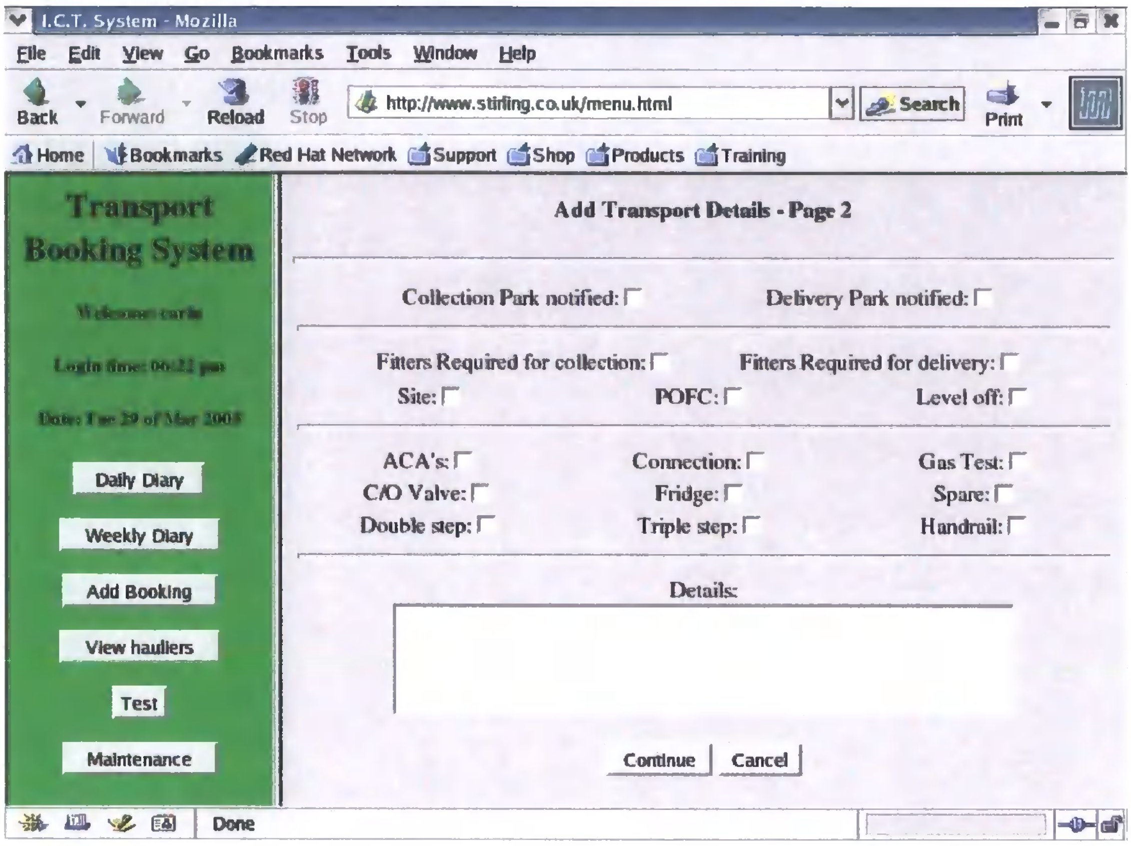
Monday	Tuesday	Wednesday	Thursday	Friday
Frank Grange	Greenlake 1	Frank Grange		Own Transport
Willerby Vogue	Atlas Festival	Atlas Nevada		Cosalt Devon
44444	123	2345		6666
Percy Woods	Denemouth	Denemouth		Seafield
Seafield	Bradford Kaims	Langley Moor		Denemouth
Frank Grange	Own Transport			
Atlas Festival	BK Charisma			
12345	333			
Bradford Kaims	Langley Moor			
Denemouth	Langley Moor			

Figure 7.2 The Weekly Diary view from the Internet based System.

The diary view as seen in Figure 7.2, whilst although basic was found to be adequate. It was loosely based upon the screen visible in Figure 4.4 which was the weekly view from BookingSystem.com. One interesting point was that users regularly printed the weekly view on a Monday and displayed it on the office wall. This occurrence could be compared with the old system where the sales director would make a copy of the physical diary and email it to all sales staff on a Monday morning. This action demonstrated the complexity of trying to introduce new operating procedures as this act had not been planned or prompted.

Management. The management involved in the testing of the system all did so at the Durham branch. It may have been more beneficial to the system for it to have been reviewed both at Morpeth and Durham, as this would have been better placed to illustrate the system's availability over the Internet but due to time constraints and for ease of testing this did not happen. The agreement, in general was that whilst the staff preferred the Internet version it offered less flexible reporting features. This was mainly due to the reporting facilities having to be all created by code, and thus not offering the type of GUI you might expect from a Window's product.

The management also noted the lack of GUI refinements which would be expected of an application; tabbing around options was difficult, screen layout wasn't perfectly aligned and lack of tool tips. These missing refinements were a result of using pure HTML to format the GUI displays. As was pointed out previously, if these issues were a real concern, they could be addressed with the introduction of a design suite such as Dream Weaver. As these were not part of the requirements, they should be classed as external to the review. It is difficult however to avoid being critical when requested for a review.


I.C.T. System - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://www.stirling.co.uk/menu.html
Search Print
Home Bookmarks Red Hat Network Support Shop Products Training
Daily Diary
Weekly Diary
Add Booking
View hauliers
Test
Maintenance
Continue Cancel
Done

Transport Booking System

Welcome carle

Login time: 06:22 pm

Date: Tue 29 of Mar 2005

Add Transport Details - Page 2

Collection Park notified: ☐ Delivery Park notified: ☐

Fitters Required for collection: ☐ Fitters Required for delivery: ☐

Site: ☐ POFC: ☐ Level off: ☐

ACA's: ☐ Connection: ☐ Gas Test: ☐

C/O Valve: ☐ Fridge: ☐ Spare: ☐

Double step: ☐ Triple step: ☐ Handrail: ☐

Details:

Figure 7.3 Page 2 of the Data Entry Screen from the Internet based System.

Project Team. The Internet system came as quite a challenge. Writing code was slow and in some cases repetitive. One of the most important points to consider was the question of whether the technology used was restrictive in any way. The main restriction was in the need to be constantly online to operate the system. The use of HTML for formatting the GUI's was also restricting, and the resultant GUI's suffered; in Figure 7.3 it is possible to see the tick boxes not lining up. A Web Development as mentioned previously would have also helped in creating dialogue boxes for certain uses; log in, prompt dialogues and printer option dialogues. After that point, the technology offered fewer restrictions or indeed more opportunities for future technological advancement within the company. One example of this would be the system's availability on PDT's.

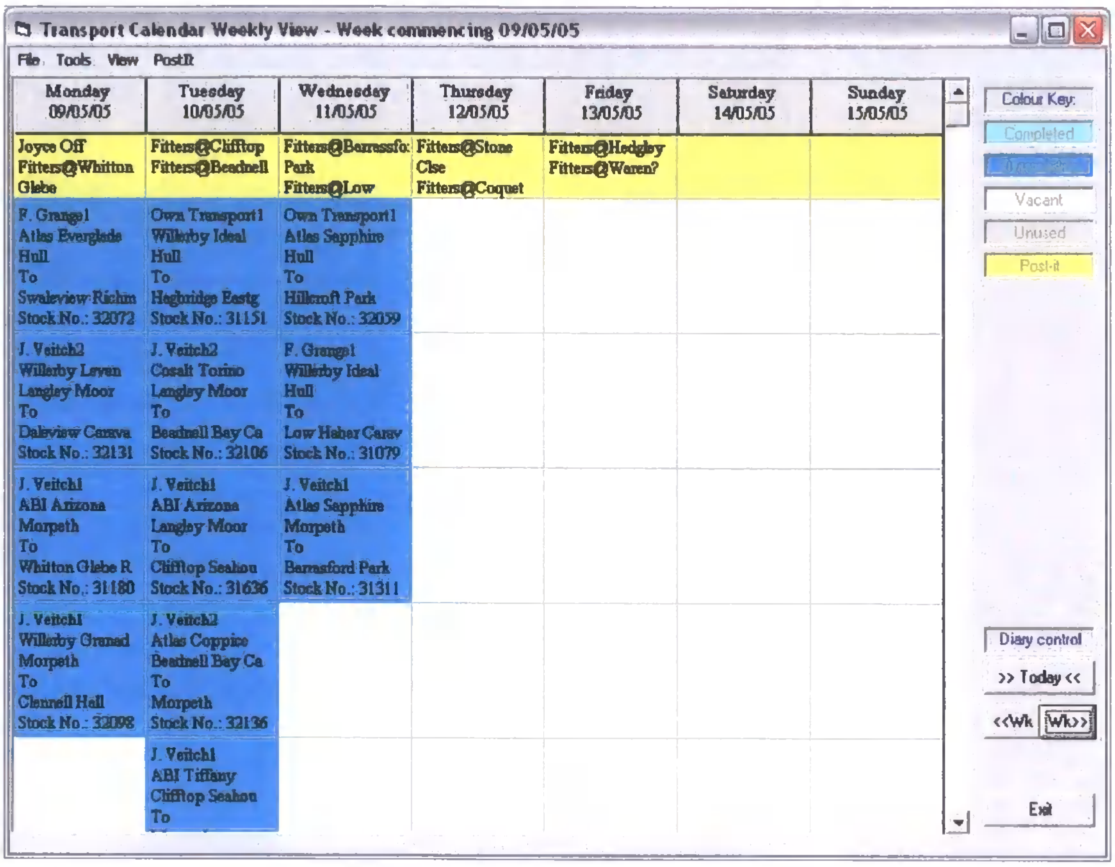
7.1.3.2.2 The Windows Based Version

The following section highlights the outcome of the evaluation of the Windows based system comparing the differing perspectives of staff and management.

Staff. The Window's system was evaluated both at Durham and Morpeth. The staff liked the diary screen and the data entry screens for clarity and its Windows feel. They liked items such as the text which pops up when the cursor is stationary on any element of the screen. They also though that having all the options available from a menu was better suited that having to click a series of buttons.

When evaluated at Morpeth over the broadband connection with all the other company systems being used, the negotiation between diary days was noticeably slow. This was due to the weekly view running a query against the database for each day of that week. Even if each query only took ¼ of a second, the seven queries could take 1 ¾ seconds. The result of this was that if a user wished to scan through one month's bookings, it may take up to 12 seconds.

The staff stated that they liked the ability to be able to use the system off-line. With this in mind, according to the log files after the first two weeks no user actually used this feature again preferring to make any changes whilst online. This feature had come with considerable overheads in development time, taking approximately 8 hours to implement.



Monday 09/05/05	Tuesday 10/05/05	Wednesday 11/05/05	Thursday 12/05/05	Friday 13/05/05	Saturday 14/05/05	Sunday 15/05/05
Joyce Off Fitters@Whitton Glebe	Fitters@Clifftop Fitters@Beadnell	Fitters@Barrassfo: Park Fitters@Low	Fitters@Stone Clse Fitters@Coquet	Fitters@Hedgley Fitters@Waren?		
F. Grange1 Atlas Everglade Hull To Swaleview Richm Stock No.: 32072	Own Transport1 Willerby Ideal Hull To Haghridge Eastg Stock No.: 31151	Own Transport1 Atlas Sapphire Hull To Hillcroft Park Stock No.: 32059				
J. Veitch2 Willerby Leven Langley Moor To Daleview Carava Stock No.: 32131	J. Veitch2 Cosalt Torino Langley Moor To Beadnell Bay Ca Stock No.: 32106	F. Grange1 Willerby Ideal Hull To Low Haber Carav Stock No.: 31079				
J. Veitch1 ABI Arizona Morpeth To Whitton Glebe R Stock No.: 31180	J. Veitch1 ABI Arizona Langley Moor To Clifftop Seahou Stock No.: 31636	J. Veitch1 Atlas Sapphire Morpeth To Barrasford Park Stock No.: 31311				
J. Veitch1 Willerby Granad Morpeth To Clennell Hall Stock No.: 32098	J. Veitch2 Atlas Coppice Beadnell Bay Ca To Morpeth Stock No.: 32136					
	J. Veitch1 ABI Tiffany Clifftop Seahou To					

Figure 7.4 The Daily Diary view of the Windows System

Another feature worth mentioning was the ability to place a Post-it at the top of the day to make a pre-booking as can be seen in Figure 7.4. This turned out to be one of the key functions of the system. In the system requirements, this feature had been classed as only desirable as opposed to essential.

Despite the above facts, and referring to Figure 7.4, the staff generally preferred the Internet version for its clarity and speed. The clarity was a result of the basic page views, and the speed was due to query speeds over the Internet.

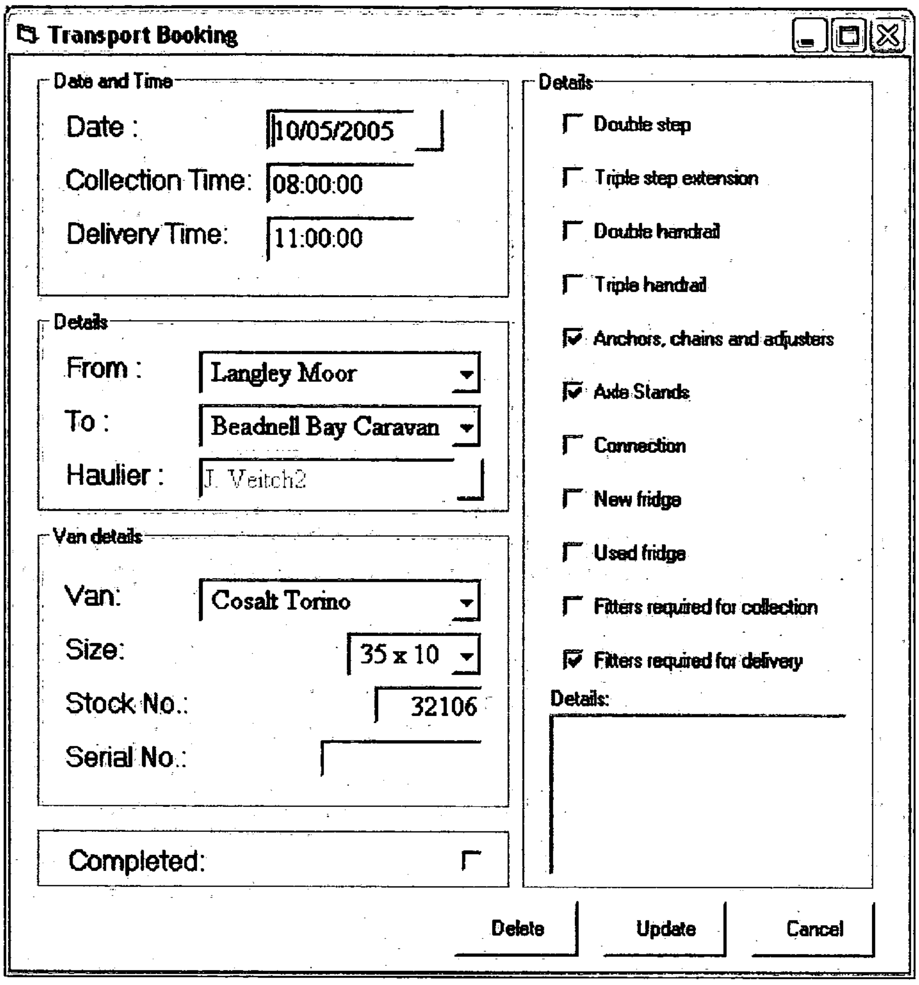
Management. The management team reviewed the system at Durham. Query times were good over the Durham network, and they felt that the system looked a lot more professional than the Internet based system; this was a result of the VB development environment offering such a professional finish. The system offered flexible reporting through VB's Reports facilities. The management liked the facility where it was possible to see which users where currently using the system. They also liked that fact that the database was Microsoft Access, which they were familiar with using alongside Excel. It was felt that the familiarity of the system's GUI was of great benefit.

One other benefit of creating the system in VB was if the system ever needed to be expanded, the company was aware of individuals who could move the system forward. They were not aware of anyone who could develop in PHP. This is an important issue for an SME.

Design Team. Of the two systems, the Windows version was visually superior. VB offers a development language and support tools which enables the speedy development of software with a highly professional look and feel. For example features such as right clicking on the tables and tool tip text were not available in the Internet version. Development speeds were also higher in VB; with reference to Figure 6.14, it took 2% more code to write the Windows version, but it took 18% less time.

The Windows version ran slower at Morpeth than did the Internet version, this was a result of the larger quantity of data that an MS Access queries returns. If this proved to be a real problem it may need to be addressed.

It was found that the company's dated laptops struggled when running the Windows system alongside the company's mail and CRM applications. This was not so apparent with the Internet version possibly due to its smaller use of system resources. However, this is only a temporary consideration as the laptops were hopefully going to be replaced by PDAs and new desktop PCs installed in the near future.


Transport Booking

Date and Time

Date : 10/05/2005

Collection Time: 08:00:00

Delivery Time: 11:00:00

Details

From : Langley Moor

To : Beadnell Bay Caravan

Haulier : J. Veitch2

Van details

Van: Cosalt Torino

Size: 35 x 10

Stock No.: 32106

Serial No.:

Completed: [ ]

Details

- [ ] Double step
- [ ] Triple step extension
- [ ] Double handrail
- [ ] Triple handrail
- [x] Anchors, chains and adjusters
- [x] Axle Stands
- [ ] Connection
- [ ] New fridge
- [ ] Used fridge
- [ ] Filters required for collection
- [x] Filters required for delivery

Details:

Figure 7.5 The Data Entry view of the Windows System

7.1.4 The Chosen System

The implementation of an Internet based version would have been appropriate considering the view that the company were intending to invest in hand-held PDAs. An Internet based version would have been available on PDAs with no alteration. It may also have lead the way to further Internet based developments which many would have considered the way forward for a company which operates in such a distributed environment.

At the Analysis stage, the company had initially suggested that only a Window's system be implemented because:

- The users were comfortable with the OS.

- Previous developments had been done using it.
- Visual Basic was supported by good reporting tools.
- The technology was known to work on the company network – this was reassuring to the company management.
- The staff knew how to query and work with Access databases.

Ultimately, the Window's version was chosen by the management at IC. This was because of the above preferences and also for the following reasons:

- The diary screen was preferred
- The offline capability was continuously liked despite its lack of utilisation.
- The speed of development
- The familiarity and the better functionality of the Windows product.

With this decision in mind, it is ironic that the staff generally preferred the Internet version. This may have been a result of the users tuning into the current trend of 'Internet-ising' everything and so being familiar with the format. It may also relate to the practice of surfing the Internet whilst the Internet browser was open, as had been done on a number of occasions. There were a number of other non-tangible reasons why some staff preferred the Internet based version, for example one user stated that it had "just had a good feel to it".

In further support of the Internet based system, it is worth referring back to Section 3.5 where it was established that those companies who were specialising in systems of this type were creating there own Internet based systems; this seems to confirm the viability of such systems.

Despite the fact that both systems shared the same core functionality, each had unique quirks relating to the associated technology and were mentioned in Chapter 5.1. These quirks sometimes impact on the user's opinion of a system. The Windows OS offers a multitude of such functions such as allowing the copying of text onto the clipboard, and the availability or right mouse click options. The Internet version also offered some of these quirks such as the ability to return to the previously viewed page by clicking the 'Back' button. This is despite the fact that the system was not coded to allow this navigation. Another function was the ability to open multiple browser pages once logged in to be able to see more than one week's bookings at any one time. It is worth considering if these sub functions had any impact on system choice.

7.1.5 The Level of Success in Correcting the Problem Domain

The main concerns with the existing method which the sales staff were using to organise the transporting of vans was that it was ad-hoc and open to misuse. There were a number of ways in which the company could have addressed this issue; a change of staff structure, staff re-training, an improved paper based system or a computer system. They chose to implement a computer system which would guide users through the correct procedure. In effect the computer system pre-defined the process and the organisation of the information relating to the transportation process. In as much as this, the implementation was a success, for instance, the system ensured that the correct data was being collected prior to the booking of movements and furthermore that the same data was available to everyone.

Point 3 of 6.3 identified an aspect of the paper based system which had been the competition between sales staff, and the manipulation of bookings enabling senior staff to be able to prioritise work to their own advantage. The computerised system addressed this successfully enabling only those with administrative privileges to make such booking changes. This was considered to be an important aspect to recruiting and retaining new staff which had previously been found problematic.

Part of the solution to obtaining a successful system was the correct identification of key points within the process. The use of Critical Success Metrics (CSMs) would have assisted in this process. CSMs are "an assessment of a running program which reflects the business concerns that prompted the creation of that program" [Menzies98]. If the CSMs can be identified correctly at the analysis stage, the chance of system success is relative to their accuracy; i.e. clear accurate CSMs equates to a successful system. It would be recommended in the CP to utilise this method of identifying success metrics as part of the proposed formal process.

Item 5 of 6.3 pointed out that there may be other less tangible benefits to introducing a computerised system, but despite the system being a successful BPR implementation, it seemed that IC problems could not all be blamed onto a lack of formalised business practice. It was evident that at least some of the blame was because of lack of staff discipline, and lack of management discipline on the staff. An example of this is the regular occurrence of sales staff arranging their day off to coincide with the transportation they have arranged; in other words to not be available if things should go wrong. This is not an aspect which can be changed by the introduction of a system, but can be changed by the company's management ensuring that those staff are available to deal with any problems that may be encountered.

The system can only assist in as much as having its users gather and enter the relevant information and to protect that information. The validity of the data would still be at the onus of the users.

8 Evaluating the Methods used in the Re-engineering the System

This section evaluates the methods used in the re-engineering process applied to the problem domain. It is organised into:

1 A description of the methods used in the re-engineering processes.

2 A set of proposals to improve the methods used.

3 The rationale behind those proposals.

8.1 A description of the methods used in the re-engineering processes

8.1.1 Analysis

The analysis stage of the project was conducted in as much isolation to IC as was possible. This meant that IC had no input into the formal analysis of the (then) current system and operation. It gave the opportunity of being critical of the current procedures without the need to take account of others who may fear the proposed changes for a variety of reasons. Whilst problems may still be present over time, currently the analysis can be deemed a success in as much as it had clarified the working domain in an understandable way.

The methods used included:

1. Identifying current roles and hierarchy with the company
2. Gathering and analysing paperwork involved in the current procedure.
3. Identifying and analysing workflows involved in the current procedure.
4. Identifying and analysing data flows involved in the current procedure.
5. Establishing the entities e.g. the bookings and the hauliers, and the functionalities e.g. the booking process of the current system.
6. Analysing the problems within the current process procedures

8.1.2 Design

The design would be based on the functional requirements stated in Figure 2.9 and the NFRs as stated in Figure 4.1.

Data Storage. Because of the collected documentation, it was known which entities needed data stored about them. This meant that the database design could be created with the knowledge that there would be very few changes expected of it as it was implemented. The availability and security of the data had also been agreed.
GUI Environment. It had been agreed that the environment should reflect that of previous company systems and so left very little doubt as to how it should be implemented. Because of the requirements, the details for the data entry and diary screen were known.
Reporting. There had been no reports associated with the paper based system, and so the required reports had been agreed as per points 9.1 and 9.2 of Table 2.1.

8.1.3 Implementation

The systems were developed using Boehm's Spiral Model [Boehm88]. This model proposes designing and implementing a system through planned iterations. The newly introduced method of tracking and controlling additional requirements is in keeping with Boehm's Spiral Model, which allows the filtering in of new requirements at each iteration. With this in mind, the size, priority and impact of each added requirement, needed to be considered before deciding whether it should be classed as a major or a minor addition. Minor alterations may be accommodated within the cycle, but more significant ones would need to be more closely planned.

Each iteration was broken down in to a number of stages:

1 Define the aims of the stage.
2 Assess the requirement criteria relating to those aims.
3 Re-engineer those requirements into code.
4 Test the code against requirements

5 Review the code against the aims of the stage
6 Make any highlighted amendments.

The following tables describe the iterations which took place during the system's development. In the tables the letter T stands for time taken in hours.

Iteration 1:

Windows Version	T.	Internet Version	T.
A basic GUI was created for the Windows version.	3	A basic framework was created which the web pages would be displayed in. A relaying system to pass information from one web page to another was put into place.	13

Table 8.1 Iteration 1

A basic GUI framework was created for the Windows version including influences from current company systems such as colours and layout.

A web page frame was created in which would sit the web pages when they were requested. During this process, it was found that CGI/C++ was unable to give the support which the application needed in regards to maintaining state between web pages. Without the writing of all system variables to an external file, they were lost from one web page to another. It was found that by using PHP which could maintain state, this problem was overcome. It was decided to halt the use of CGI/C++ and create the dynamic parts of the web page in PHP. The result of this was that 9 hours were wasted on CGI/C++ coding which was later discarded. Those 9 hours have been included in the time taken figure of 13 hours.

Iteration 2:

Windows Version	T.	Internet Version	T.
The menu system was put into place and agreed after testing. Login functionality was added.	3	A basic screen layout within the frames was created in HTML. The login process was created as part of this utilising the relaying system from iteration 1.	6

Table 8.2 Iteration 2

The VB menu wizard was used to put the menu system in place on the Windows version. Due to HTML restrictions, buttons would be used on the Internet version for this purpose. Utilising the message relaying system, a log in function was created for the Internet based version. Once this feature was completed, each web page was capable of checking if the user was currently logged in. The VB version only needed to ensure the user logged in at the beginning of the program when the system would be first executed.

Iteration 3:

Windows Version	T.	Internet Version	T.
The database was created, and filled with test data. Database links were created to VB	4	The database was created, and filled with test data. Database links were created to PHP	4

Table 8.3 Iteration 3

The databases used for both systems were identical. The time used to design the database which was 2 hours was distributed equally to the two systems. The actual creation of the two databases took only 1 hour each. The remaining time was used to create the links between the program and the database.

Iteration 4:

Windows Version	T.	Internet Version	T.
Data entry and diary screen were mocked up for appraisal	7	Data entry and menu screen were mocked up for appraisal	5

Table 8.4 Iteration 4

In this iteration, the data entry screen and the diary views were created. The Internet version of the diary view utilised a recursive table arrangement and was quite simplistic in its approach. The Windows version was more complex needing to implement a specialised table which comes as part of the VB package.

The data entry screens in both cases were very similar with regard to the placing of the button and boxes onto the screen and the corresponding code relating to them.

Iteration 5 +:

Windows Version	T.	Internet Version	T.
The main areas of functionality was added and evaluated.	26	The main areas of functionality was added and evaluated.	32

Table 8.5 Iteration 5+

This was by far the lengthiest iteration and because of this, benefited from being broken down into a number of lesser iterations each one concentrating on an area of functionality. The lengths of cycles were dependent upon the complexity of the particular functions. This fact together with the need for differing functions made the process of creating the two systems in parallel, difficult. The Windows version could be created in a more modular fashion due to the ability of being able to use object-oriented development approaches within VB. With the Internet version, development was more along the lines of processing web pages in steps. Because of the higher level of fragmentation and the more coding required it took approximately 6 hours longer for completion.

The development of the two systems in parallel using iteration was found to be difficult in practise for the following reasons:

- With VB, the basic GUI environment can be created rapidly due to its collection of development tools. The code could be written for each object of the GUI in a logical order, some possibly being done much later in the development. Functionality can be added in any suitable order. This method of development is in keeping with Boehm's Spiral Model. For the Internet version PHP was used for scripting, and unfortunately this meant that the code was more disjointed in as much as each viewed page could be perceived as a stand alone mini-application, and consequently was more oriented by the functionality of that singular Internet page than the functionality of the process. There were a number of global functions, but again each had to be implemented as a separate scripted page. This made the process of creating the two systems in parallel more difficult.
- Whilst expertise in VB was available in IC, there was no expertise available for the development of web pages. This had an impact on making the development process slower. Specifically, experimentation was required to ensure aesthetics of the web pages as well as other factors such as security, platform issues and set up.
- Each web page needs a script within it to offer any level interaction. Unfortunately, as each web page and therefore each script is executed independently of one another, there is method of sending a message, in this case a variable, directly from one web page to another. This obstacle may be overcome in a number of ways within the PHP coding. The result of this is an increase in the development time.
- Writing scripted web pages require a number of linked individual programs, much as in a linked-list approach. Each web page is a small program in itself. In this case, for each web page to be successful it must:

  1 Ensure the user is logged in.

  2 Import the necessary data, possibly from the previous screens.

  3 Process and display the data.

  4 Record the data again for the next time it is needed.

  An example of this in use would be when the user negotiates through different days in the diary view; each time, the currently viewed date must be passed along to the next page. When this method is compared against the VB equivalent which simply remembers the date, it is possible to see why the time taken to complete each iteration is much longer in the Internet based system than the Windows one.

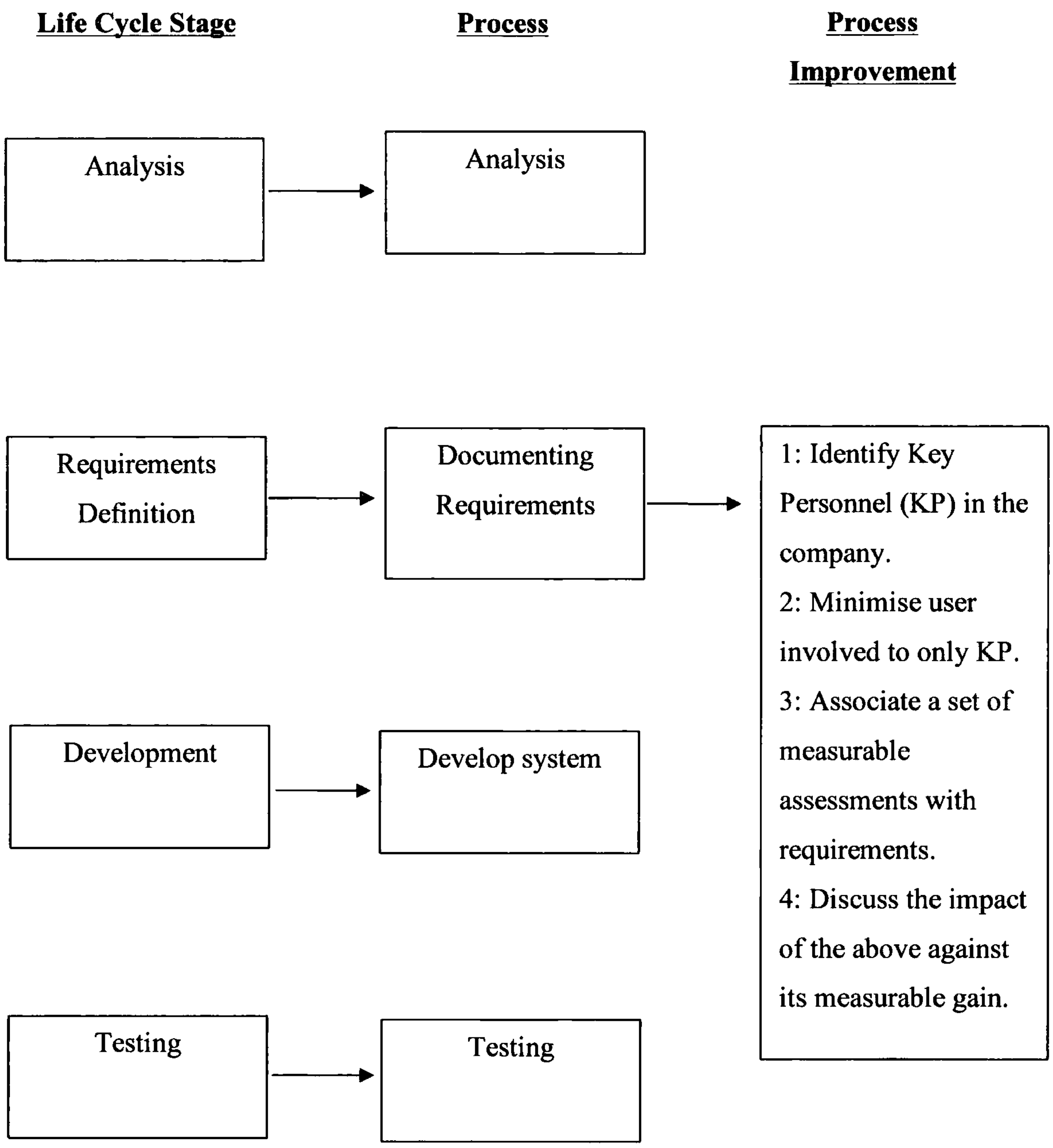
8.2 A Set of Proposals to Improve the Methods used

Boehm's Spiral Model strongly promotes the application of project plans. IC arrangements with regard to the setting of deadline typically lacked detail. Informal agreements had been made about when the project should start, when it was expected to be evaluated and when it was to be installed. The following section illustrates the proposed changes to IC methods of systems re-engineering.

To illustrate these changes, diagrams have been used. The diagrams have 3 columns comprising:

1 The Life Cycle Stages
2 The Processes, of which one is to be changed
3 Details of the process improvement.

The first proposal was a change to the method by which the requirement definitions were collected.



Life Cycle Stage
Process
Process Improvement
Analysis
Analysis
Requirements Definition
Documenting Requirements
1: Identify Key Personnel (KP) in the company.
2: Minimise user involved to only KP.
3: Associate a set of measurable assessments with requirements.
4: Discuss the impact of the above against its measurable gain.
Development
Develop system
Testing
Testing

Figure 8.1 Process Improvement 1. Design Stage – Optimising the Requirements Collection

The change would encompass the following points:

1 Identifying key personnel (KP) who would represent each body of the user community. Only these KP would have direct input during into the project. Input from other member of staff should be channeled through their respective representative who can consider and raise the issues in needed. These KP should be in a position where they thoroughly understand the processes which are to be re-engineered within their specific business area. This should control the amount and quality of requirements suggested.

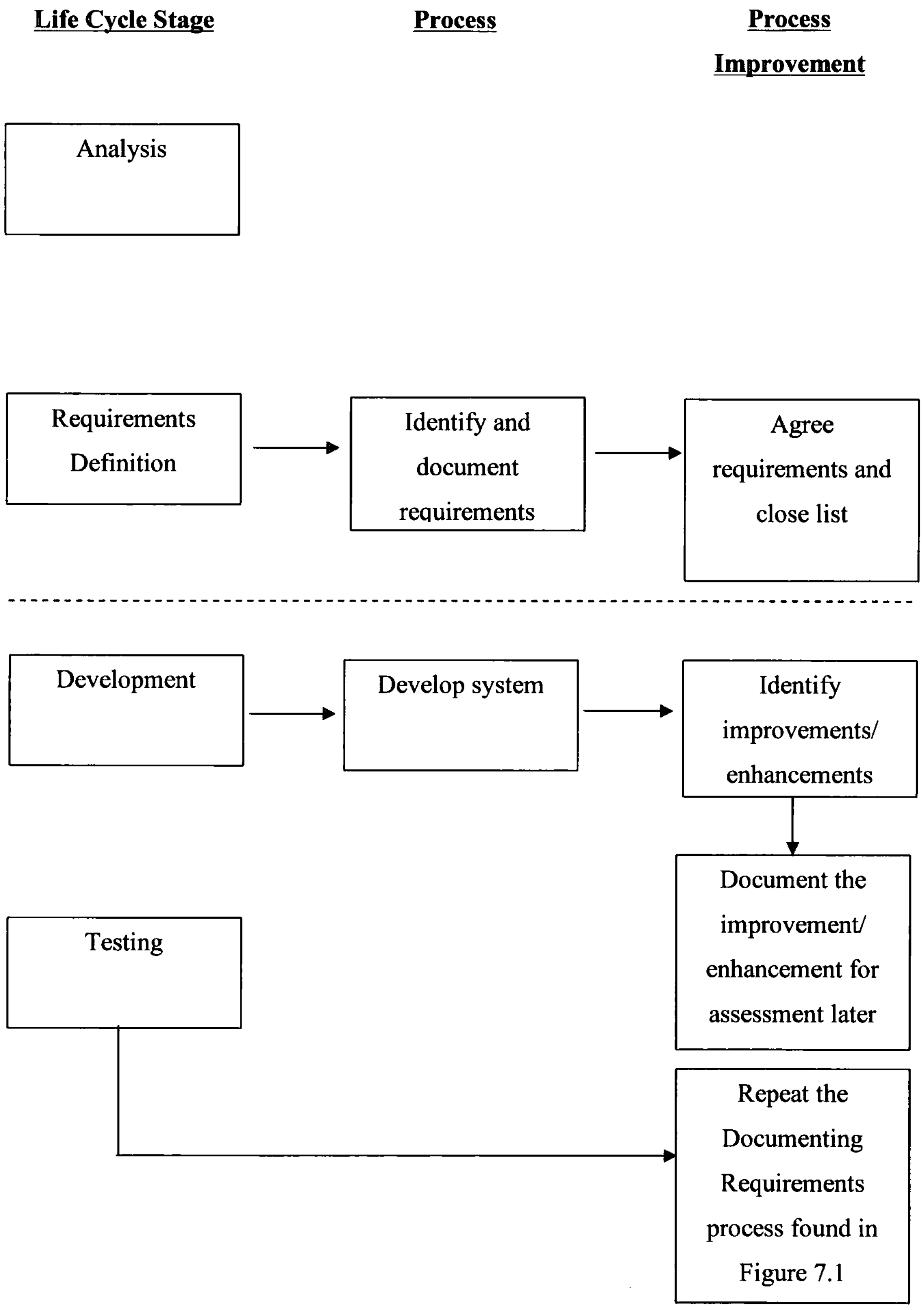
2 Associate a set of measurable assessments with requirements.
Associate a measurable assessment against a suggested requirement. This measurement or measurements may be different from system to system, but should be agreed before commencement. Examples of measurements are:

   A Development time.
   B Potential business advantage.
   C Expanding the systems availability to users.

3 Discuss the above with its measurable gain.
The gain from implementing the requirement should be considered against the time that would be needed in creating it and other issues such as the impact it would have on other areas of the system, such as; might the life of the system be lengthened or shortened.

This change was introduced as a result of the quantity and quality of user input. Whilst input from the user and benefactor are always relevant, the manner in which the design process was implemented led to numerous, highly complex and sometimes conflicting requirements. It was considered a possibility that KP would be able to develop their skills in this process over a number of projects in the future.

The second proposal was to introduce a change management procedure.



Life Cycle Stage
Process
Process Improvement
Analysis
Requirements Definition
Identify and document requirements
Agree requirements and close list
Development
Develop system
Identify improvements/ enhancements
Document the improvement/ enhancement for assessment later
Testing
Repeat the Documenting Requirements process found in Figure 7.1

Figure 8.2 Process Improvement 2. The Design Stage – Change Management

This change would involve the following points:

1 Identify improvements and enhancements. Improvements and new enhancements may be identified at any stage of the development process, some may have been carried over from the requirements stage.

2 Agree requirements and close list. Once the requirements list had been agreed, no more requirements should be added (until the next iteration).

3 Document the improvements/ enhancements for assessment at a later phase. The improvements/ enhancements will be considered as in 8.2 at a later phase. Note that they are not authorised as requirements until after this has been done.

4 Repeat the Documenting Requirements process found in 8.2
Once the 1st prototype of the system has been successful, the outstanding improvements/ enhancements can be considered.

This improved process was introduced due to the requirements creep experienced. Prior to this change, the design process and the development time was longer than had been planned; designing the system had been scheduled for a two week period, but ended up taking 4 weeks (pro-rata).

When the design of the system was agreed, any major changes should have been documented for later consideration instead of being included. Prior to this, the company insisted that changes should be included in the 1st prototype of the systems. See Table 8.6 for details of how the additional requirements amounted to almost 13% of the total coding time.

IC management placed pressure on the development team to complete the design within a very short period of time. This can be expected in a company where the general expertise and understanding of computer systems is limited. If a new function was to be omitted then expectations would need to be managed as failure to incorporate the proposed revisions may mean that the staff consider the work to be only a partial success. This accounted for the requirement creep, and the consequence highlighted the importance of strict control of requirements and demonstrated a strong need for a requirements change management process. This proposed change is represented in Figure 7.2 which shows how after the initial requirements list has been produced; no further requirements may be added to the first release of the system. Any new requirements should be documented. New requirements may be included in the system at a given point or after the completion of the system (in the second release) based on assessment of the change impact and their priority. This will prevent creep and enable development time scales to be estimated.

A proposal was created for the development stage; this proposal centered on the identification and setting of boundaries of each iterative cycle.

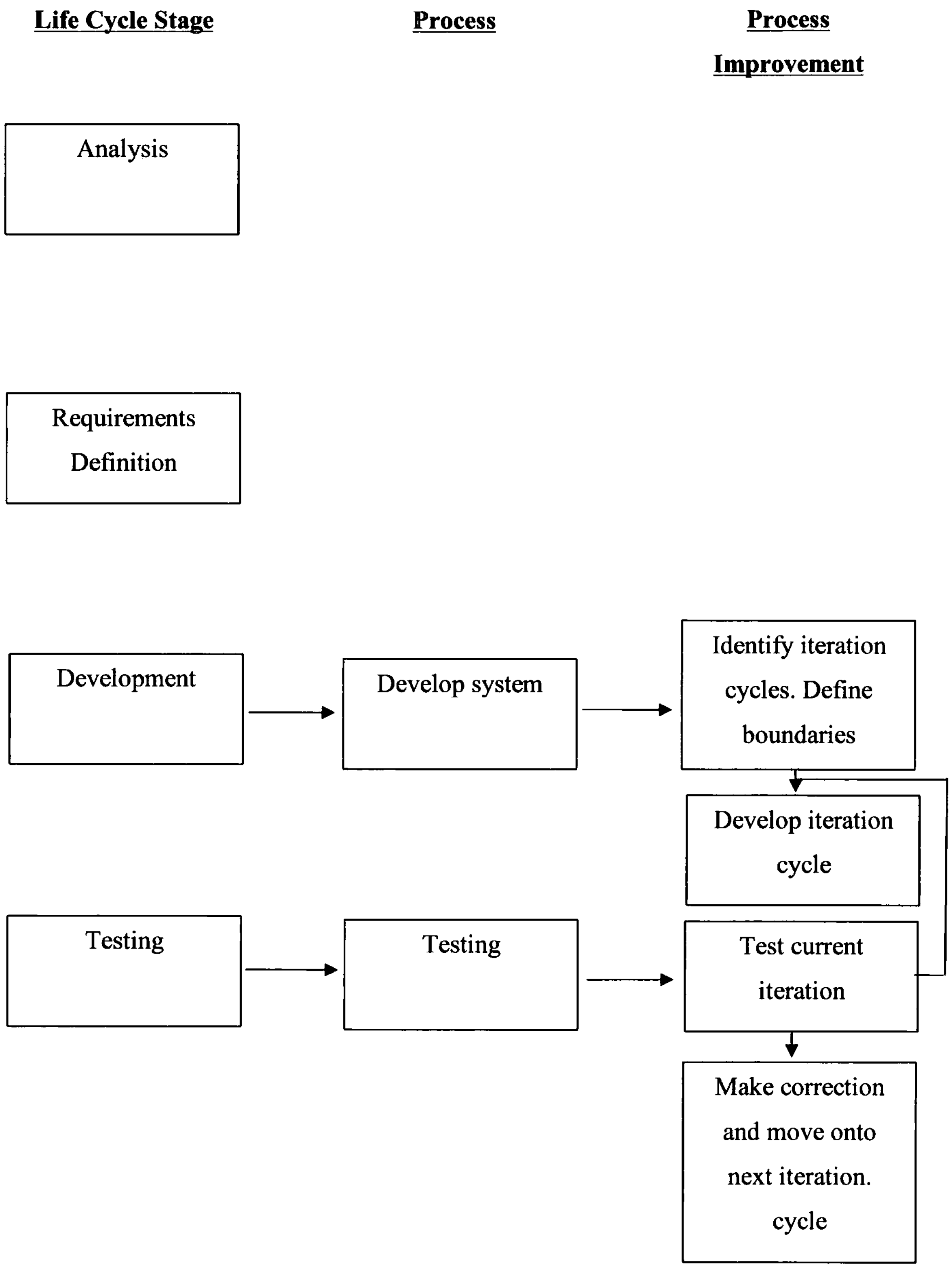


Figure 8.3 Process Improvement 3. The Implementation Stage – Iteration control

This change would involve the following steps:

1 Identify the iteration cycles and their boundaries.

Identify the main iteration cycles. Once identified, the boundaries of those cycles should be documented. Boundaries might include a time period, the data to be processed, a particular process flow etc.

2 Develop the iteration cycle

Engineer the software of the highlighted iteration based on the above boundaries

3 Test the current iteration

Execute and test the above software.

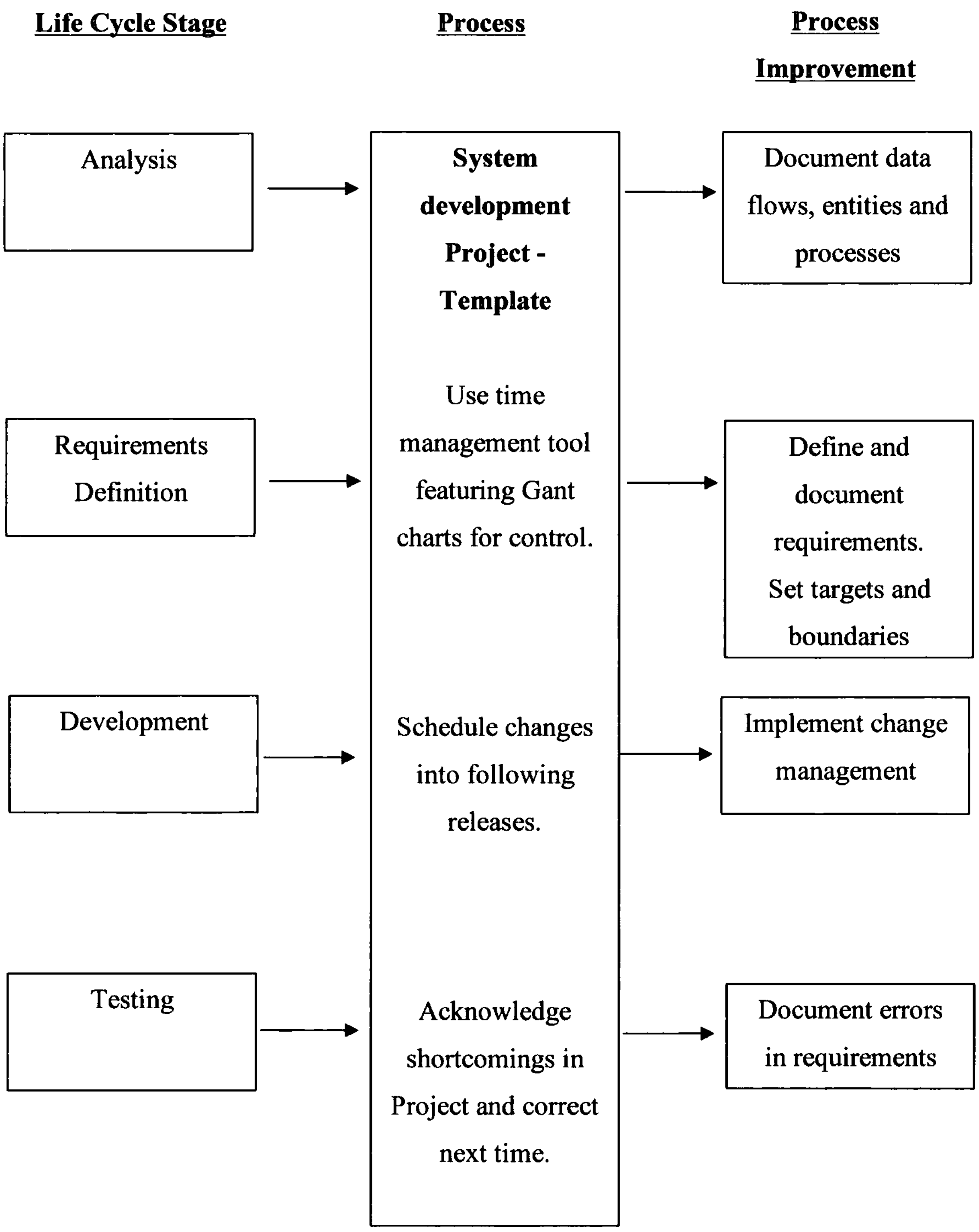
4 Make correction and move onto next iteration.

Make corrections or additions to the iteration to bring the iteration up to the identified boundaries.

As mentioned previously, the project suffered from requirements creep, and a clearly established set of requirements would help in stopping this from happening. The company had placed the developers under pressure to make additional changes in an ad-hoc manner once these cycles where under way. This resulted not only in requirements creep but in time delays as a consequence. The additional time taken to implement these ad-hoc additions averaged 12 hours between the two systems. Table 8.6 documents this problem.

Hughes [Hughes94] suggests the introduction of process improvement iteration controls as an aid to counteract requirements creep. As part of this, the use of tools such as JXProject and MS Project to plan the time scales and dependencies should be utilised. It is difficult to assess if a project is over-running if no time scales are in place. It is also difficult in a small business to adhere to procedures and timetables when there is management pressure to add in extra functionality, but with better planning, more rational arrangement of their prioritisation can be made.

The final process improvement centered upon the creation of a template from which the company could base future engineering projects.



Life Cycle Stage
Process
Process Improvement
Analysis
Requirements Definition
Development
Testing
System development Project - Template
Use time management tool featuring Gant charts for control.
Schedule changes into following releases.
Acknowledge shortcomings in Project and correct next time.
Document data flows, entities and processes
Define and document requirements. Set targets and boundaries
Implement change management
Document errors in requirements

Figure 8.4 Process Improvement 4. The Project Template

The process improvement strategy consisted of the creation of a template which would help the user to:

1. Improve performance during the Analysis stage; correctly identify data flows, entities and processes. Once identified, they need to be documented.
2. Define and document requirements, set targets and boundaries; establish requirements by examining the current domain, the proposed system, and user input.
3. Implement change management; put into place a change management plan which will control requirements and development time planning.
4. Retrospectively document errors in requirements; misleading or erroneous requirements should be documented for consideration in future projects.
5. Iteratively review the entire SE process and make changes accordingly.

The proposed solution to this problem is a guide for IC to use during future software developments. This template could be used in all future developments. Figure 8.4 illustrates how the introduction of this improvement could be managed.
By implementing the template, IC would be incorporating those improvements described above, and would be able to fully document the software engineering project in a manner which instils good working practises. An example of this is at the analysis stage; the template requests that the user itemise all data requirements for all system variables. This approach will later aid the system's maintenance. By this simple piece of housekeeping, no data will be overlooked in the current process when reengineered into the new system.

8.3 A Review of the Collected Metrics

The concept of metrics was introduced in Section 5.3 and was identified as a means to provide a method by which the two systems could be compared. The metrics allowed comparisons of the more tangible aspects of software development such as software development time. However, one issue of concern with regard to such metrics is whether those collected are those of greatest relevance. Further, there is also an issue of the accuracy of the data collection process.

Having raised such issues, consideration must be made upon how to address them; if the process of identification and collection of the metrics were being done by a specialist team, it would be viable for them to iteratively question how and what was collected over numerous projects. But in IC situation this process was going to be done by their own staff who may not have the time or capability to re-assess the necessary skills. It is worth noting that the comparison of 2 systems allowed this to be done for the 1st time at IC, and consequently some of the metrics collected may be found to be unnecessary.

With reference to Section 5.3, the collected metrics were collated and entered into Table 8.6.

[Ward01] Ward, R.P., Fayed, M.E., Laitinen, M., Software Process Improvement in the Small, Communications of the ACM, Vol. 44 No. 4, pp105, 2001

[Wasserman80]Wasserman, A.I., Information system Design Methodology. Journal of the American Society for Information Science, Jan. 1980

[Wiegers99] Wiegers, K.E., Why is Process Improvement so Hard?, Software Development Online, 1999

[Zahran98] Zahran S., Software Process Improvement – Practical Guideline for Business Success, Addison-Wesley 1998

[Zhao98] Zhao, J., On Assessing the Complexities of Software Architectures, Proceedings of the 3rd International Software Architecture Workshop (ISAW3), pp.163-166, ACM SIGSOFT, November 1998

[SEIR06] The Software Engineering Institute Repository, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, HTTPS://seir.sei.cmu.edu/seir, [Accessed 12/11/06]

[Shum93] Shum, S., QOC Design Rationale Retrieval: A Cognitive Task Analysis, & Design Implications, Technical Report, Rank Xerox EuroPARC, Cambridge, UK, 1993

[Singh93] Singh R., International Standard ISO/IEC 12207 Software Life Cycle Processes, Federal White Paper, 1993

[Sommerville95]Sommerville I., Software Engineering, Addison-Wesley Longman Ltd., Harlow UK. 1995

[Sommerville97]Sommerville I., Requirements Engineering, John Wiley and Sons, Chichester UK, 1997

[SPICE06] SPICE: Centre for IT Research and Technology Transfer (ITC) University of Boras, SE-501 90 Boras, Sweden, http://www.sqi.gu.edu.au/spice/, [Accessed 11/10/06]

[SPIN06] The Ottawa SPIN, Queensview Drive, Suite 200 Ottawa, Ontario, Canada, http://www.spin.org/ [Accessed 15/06/06]

[SPIRE106] Software Process Improvement in Regions of Europe (SPIRE) - ESSI project 23873, [accessed at http://www.cse.dcu.ie/cse_www/publications.htm on 15/11/06]

[SPIRE206] SPIRE: Center for Software Engineering, Dublin City University Campus, Dublin, Ireland, http://www.csu.dcu.ie/spire/, [Accessed 11/10/06]

[Trillium94] Unknown, Trillium Software Data Quality Methodology, White Paper, Trillium Software, 1994. [accessed at http://www.trilliumsoftware.com/site/content/resources/library/pdf_detail.asp?ID=163 on 15/11/06]

[TSL06] Texas State Library and Archives Commission, Available from: http://www.tsl.state.tx.us/ld/pubs/compsecurity/glossary.html, [Accessed 20/07/06]

[Ubhayakar03]Ubhayakar, S., Evaluation of Program Specification and Verification Systems, Master Thesis, Naval post Graduate school, Monterey, California, Jun 2003

[McGraw04] McGraw, G. Risk Analysis in Software Design. IEEE Computer Society. May/June2004

[McKerlie93] McKerlie, D., MacLean, A., QOC in Action: Using Design Rationale to Support Design, Interchi, Conference and Proceedings, 1993

[Meyer03] Meyer, B., The Power of Abstraction, Reuse and Simplicity: An Object Oriented Library for Event Driven Design, Springer Verlag, LNCS 2635, 2003

[Microsoft06] Microsoft Corporation, http://www.microsoft.com/, http://www.sqi.gu.edu.au/spice/, [Accessed 11/10/06]

[Mylopoulos00]Mylopoulos, J. et al., Tropos: A Framework for Requirements-Driven Software Development, Lecture Notes in Computer Science. Springer-Verlag, 2000

[Nogueira98] Nogueira, C. and Luqi. A Formal Risk Assessment Model for Software Evolution. White Paper, Naval Postgraduate School. 1998

[OSC06] The Open Source Organisation, 2006, Available from: http://www.opensource.org/, [Accessed 06/07/06]

[Paulk94] Paulk M., Weber C., Curtis B., Chrissis M.B., The Capability Maturity Model – Guidelines for improving the Software Process. Addison-Wesley, 1994

[Rifkin02] Rifkin, S., Is Process Improvement Irrelevant to Produce New Era Software, Proc. Software Quality: Quality Connection, pp13-16, Springer-Verlag, Germany, 2002

[Rosen67] Rosen, S., Programming systems and languages--a historical survey. In Programming systems and Languages, McGraw-Hill, New York, 1967

[Ross95] Ross, S., Ramage, M., Rogers, Y., PETRA: Participatory Evaluation through Redesign and Analysis, Collaborative Research Support Program, No. 375, 1995

[Royce87] Royce, W., Managing the Development of Large Software Systems: Concepts and Techniques, WESCON, August 1970; reprinted in Ninth International Conference on Software Engineering, IEEE Computer Society Press 1987, pp 328-338

[Saukkonen01]Saukkonen S., Software Process Improvement Research Action Laboratory, Annual Report, University of Oulu, Finland, 2001

[SEI06] Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, Available from http://www.sei.cmu.edu/ [Accessed 11/10/06] http://www.iso.org/iso/en/xsite/contact/contact.html, [Accessed 11/10/06]

[Kumar90] Kumar, K., Post Implementation Evaluation of Computer-based Information Systems: Current Practices, Communications of the ACM, Volume 33, Issue 2, February 1990

[Kemerer96] Kemerer, C. Reliability of Function Point Measurements. Communications ACM, Vol. 36 No 2. 1996

[Kettinger97] Kettinger, J., Teng, J., Guha, S., Business Process Change: A Study of Methodologies, Techiques and Tools, MIS Quarterly, Vol. 21 No. 1, 1997

[Konti95] Konti, J., Chen S., Limperos, K., Tesoriero, R., Caldiera, G., Deutsch, M., A COTS Selection Method and Experiences of its Use, Presented at the NASA Software Engineering Laboratory, 1995

[Landis92] Landis L. et al., Recommended Approach to Software Development, NASA Software Engineering Laboratory Series, Revision 3, 1992

[Lin1] Author unknown, 2006, Article 7564, Linux Journal, www.linuxjournal.com/articles/7564, [Accessed 02/09/06]

[Lin2] Author unknown, Linux Magazine, www.linux-magazine.com, [Accessed 02/09/06]

[Malhotra00] Malhotra, Y. Knowledge Management and New Organisation Forms: A Framework for Business Model Innovation, Knowledge Management and Virtual Organisations, Hershey, PA: Idea Group Publishing, 2000

[Menzies98] Menzies, T., Evaluation Issues with Critical Success Metrics, Proceedings of the Eleventh Workshop on Knowledge, Banff USA, 1998

[McCall77] McCall, J., Richards, P., Walters, G., Factors in Software Quality Volumes 1 – 3, General Electric, Co., Rep. GE-TIS-77 CIS 02, 1977

[McColl01] McColl E., Jacoby A., Thomas L., Soutter J., Bamford C., Sten N., et al. Design and use of questionnaires; a review of best practises applicable to surveys of health service staff and patients Health Technol Assessment 2001

[McFeely96] McFeely, B., IDEAL: A User's Guide for Software Process Improvement, Pittsburgh, SEI Handbook, 1996

A References

[Anderson99] Anderson, M., Sohal, A.S., A Study of the Relationship between Quality Management Practices and Performances in Small Businesses, International Journal of Quality & Reliability Management, Vol. 16 No. 9, 1999

[Basili85] Basili, V.R., "Measuring the Software Process and Product: Lessons Learned in the SEL, in Proc. Tenth Annual Software Engineering Workshop, NASA Goddard Space Flight Center, Greenbelt MD 20771, December 1985

[Beecham03] Beecham S., Hall T., Rainer A., Software Process Improvement Problems in Twelve Software Companies: An Empirical Analysis, Empirical Software Engineering Vol. 8, Netherlands, 2003

[Berard06] Berard E.V., Metrics for Object-Oriented Software Engineering, Article, The Institute of Computer Science at The Polish Academy of Sciences, Undated [Access 21/09/06]

[Berners-Lee06]Berners-Lee, Tim, 2006. The World Wide Internet Consortium [online]. Available from: www.w3.org/Consortium/, [Accessed 12/04/06]

[Biro98] Biro M., Remzso T, Business Motivations for Software Process Improvement, ERCIM News, Iss. 32, 1998

[Boehm88] Boehm, B. A Spiral Model of Software Development and Enhancement. Computer. May 1988

[Boehm89] Boehm, B.W., Software Risk Management (Tutorial), IEEE Computer Society Press, New York, 1989

[Briand96] Briand, L., Differding, M., Rombach, D., Practical Guidelines for Measurement Based Process Improvement, A Technical Report of the International Software Engineering Network (ISERN), May 1996

[Briand02] Briand, L., Wust, J., The Impact of Design Properties on Development Costs In Object Oriented Systems, IEEE Transactions on Software Engineering, 2002

[Conradi95] Conradi R., Liu C, Process Modelling Langauges: One or many, EWSPT '95, Leiden Netherlands, 1995

[Conradi02] Conradi, H., Fugetta, A., Improving Software Process Improvement, Software IEEE, Vol. 19 Iss. 4, 2002

[Daft99] Daft, R.L., management, 4th Ed., The Dryden Press, U.S.A, 1999

[Davenport93] Davenport, T.H., Short, J.E., The New Industrial Engineering:Information Technology and Business Process Redesign, Software Re-engineering, IEEE Computer Society Press, 1993

[Diaz97] Diaz, M., Sligo, J., How Software Process Improvement Helped Motorola, IEEE Software, Computer Society Press volume 14 issue 5, 1997

[Doublait97] Doublait, S., Standard reuse practices: many myths vs. a reality, ACM Press, Vol. 6 No. 2, 1997

[Ebert96] Ebert, A., Classification Techniques for Metric Based Software Development, Software Quality Journal, pp255-272, Vol. 5, No. 4, 1996

[Enam98] Enam K., Goldenson D., McCurley J., Herbsleb J., Success or Failure? Modeling the Likelihood of Software Process Improvement, International Software Engineering Research Network ISERN 98- 15, USA, 1998

[ESI06] The European Software Institute, Parque Techologico de Zaudio, Bizkaia, Spain, http://www.esi.es/index.php, [Accessed 11/10/06]

[Fenton00] Fenton, N.E.,Neil, M., Software Metrics: Roadmap, Proceedings of the Conference on The Future of Software Engineering, Limerick Ireland, 2000

[Foddy94] Foddy, W., Constructing Questions for Interviews and Questionnaires: Theory and Practice in Social Research, Cambridge University Press, 1994

[Fuggetta00] Fuggetta A., Software Process: A Roadmap, Proceedings of the Conference on the Future of Software Engineering, International Conference on Software Engineering, Limerick Ireland 2000

[Ferguson99] Ferguson P., Leman G., Perini p., Renner S., Sechagiri G., Software Process Improvement Works!, Technical Report CMU/SEI-99-TR-027, Carnegie Mellon Software Engineering Institute, 1999

[Glass98] Glass, R. L. Software Runaways: Lessons Learned from Massive Software Project Failures. Upper Saddle River, NJ: Prentice-Hall, 1998

[Goransson86] Goransson, B., The Interface is Often Not the Problem, Conference on Human Factors in Information systems, Proceedings of the SIGCHI/GI

Conference on Human Factors in Computing systems an d Graphics Interface, ACM Press, 1986

[Grover95] Grover, V., Kettinger, W., Business Process Change: Re-engineering Concepts, Methods and Technologies, Ideal Group Publishing, 1995

[Haley96] Haley, T.J., Software Process Improvement at Raytheon, IEEE Software, Computer Society Press volume 13 issue 6, 1996

[Hammer90] Hammer, M., Re-engineering Work: Don't automate, obliterate, Harvard Business Review, July Issue 1990

[Hammer93] Hammer, M., Champy, J., Reengineering the Corporation, Harper Business, New York, 1993

[Hammer01] Hammer, M., Champy, J., Re-engineering the Corporation, Harper Business Press, 2001

[Harrison87] Harrison, W., Cook, C., A Micro/Macro Measure of Software Complexity, Journal of Systems and Software, pp213-219, Vol. 7, No. 2, 1987

[Heath06] Heath, Sue, 2001. Trends: eBusiness Strategy - Measuring the Future. Available from: http://www.cioinsight.com/article2/0,1540,1458593,00.asp, [Accessed 26/06/06]

[Henry81] Henry, S., Kafura, D., Software Structure Measures Based on Information Flow, IEEE Transactions on Software Engineering, pp 510 – 518, Vol. 7, No. 5, 1981

[Herbsleb94] Herbsleb J., Carleton A., Rozum J., Siegal J., Zubrow D., Benefits of CMM Based Software Process Improvement: Initial Results, Technical Report SEI-94-TR-13, Carnegie Mellon Software Engineering Institute, 1994

[Hughes94] Hughes J., King V., Rodden T., Anderson H., Moving out from the Control Room: Ethnography in System Design, Proceedings of the 1996 ACM on CSCW, Chapel Hill, US, 1994

[Humphrey88] Humphrey W., Characterizing the Software Process: A Maturity Framework, IEEE Software, 5(2) March 1988

[ISO06] International Standards Organisation, 1, rue de Varembé, Case postale 56 CH-1211 Geneva 20, Switzerland,

9.2.2 To Conduct an Informal Experimental Evaluation of the above Process

Two systems had been created in tandem, partly because IC could not decide on the platform which to base them on, and partly as an opportunity for an experiment to assist in the creation of a structured modus operandi for this dissertation work. From a practical point of view, there are not many situations would enable such an experiment and therefore the research outcomes are potentially highly valuable to the research community.

The fact that IC could not decide on a system indicated their lack of understanding in what they wanted from the system, and it also could be related to the reason why they had been the recipients of systems in the past which they felt were not entirely suitable. The systems were developed using Boehm's Spiral method and this method compliments prototyping. In hindsight, it may not have been necessary to create two whole new systems; at the point when the GUIs had been created and the functionality had been clearly defined, a decision should have been made as to which version the company wished to implement. This would have shortened production time and may have allowed the designers to focus more clearly on the one system, rather that continually referring to the other version of the system to think how it would be recreated functionally. The benefit of taking both systems to completion did however enable a full set of metrics for each system to be collated and therefore enabled detailed comparisons to be made and the realities of IC initial decision making to become apparent.

9.2.3 To Create a Structured Business Method of Software Engineering focused on the Requirements of the Company and based on the Findings of the Above

The new CP was welcomed by IC, as was the framework. It is not apparent at the time of writing if these have been a success, as the company have not had the opportunity to utilise them. Arguably, a procedure is only as good as the user of the procedure, and his/her understanding of it. Another aspect to this is difficulty in the measuring of the success of the application of a method in software engineering [Anderson99]. In other forms of engineering such as civil engineering, it is easier to see the results of the method as they are more tangible. With this in mind, it would take a suitably experienced person to be able to implement the framework on future developments and to be able to judge its benefits.

Another aspect of this is the use of the framework in conjunction with an external party in creating a piece of software, i.e. a software development company who may already have their own framework in place. In this case, IC would at least be able to use the initial sections that organise the collection of relating paperwork, process flows, entities and requirements. This would at least help to ensure that their illustration of the required system was accurate and clearly defined. As IC had pointed out that previously, they felt that the systems that had been created for them had been not wholly suitable; they felt this would be hugely beneficial.

9.3 Results

The following results list the findings made through the research and work done in this thesis:

1 This thesis has demonstrated that by the collecting of metrics throughout the software engineering process, original decision making plans could be reconsidered and appraised retrospectively. This can be viewed as a form of process improvement as the appraisal highlighted certain aspects of the original decisions which could be altered the next time ICs needed to make such decisions.

2 By implementing two systems in parallel, and the collection of metrics, it was possible to make comparisons of the methods, tools and decisions of both systems and highlight what was and was not beneficial. The outcomes are also a demonstration of a form of process improvement.

3 It was found that due to human nature and the competitive working environment of the staff, it is not always possible to purely implement a process and expect the process to be adhered to; it is hard to change established working patterns. In the case of IC's, they had requested a process improvement to their working practise, but ultimately what they needed was an enforced improved process. It

would be interesting to see if in the future, the staff at IC would use the system as it was designed or might they find ways around the enforced processes.

4 This thesis demonstrated the importance of risk management. Through the collection of metrics, it is possible to conclude that unfamiliar languages present higher risk with regard to development time than do known languages.

5 The work highlighted some of the complexities of the field of ethnography. It is also possible to establish that there is a considerable 'unknown' factor when it comes to user input into the software design lifecycle. It was found that when observing working patterns, those user when being observed acted differently to when being not being observed. This might lead to the collection of false data and data flows. This phenomenon which may be termed as ethnography was also in evidence in the questionnaire where the system which had achieved the lowest score was chosen as the system which was to be implemented.

9.4 Further Work

Although this thesis has offered only a preliminary study into software process improvements, there have been a number of areas where the research could be strengthened;

1 During the Case study, QOC was used to help in the decision making process. One of the problems encountered was that criteria differed between some options. For example when considering which programming language to use in Unix, it was found that some of the criteria for considering Cold fusion were different than some of those in PHP; Cold fusion had its own database, whereas a suitable database would need to be sourced for PHP. Unfortunately, using the QOC tables, this was not easily highlighted. Another problem encountered with using QOC was that whilst considering criteria, it was found that where some options achieved one particular criterion, it was at the expense of another. Using Cold Fusion as an example, it offered its own internal database, but at the cost of offering less functionality of say, PHP combined with the MySQL database.

It would be therefore beneficial for more research to have been done into offering a more dynamic method of illustrating these linked criteria. One such method is offered by Shum [Shum93] who suggested the use of criterion trees and note-cards. One possible extension of this approach would be to make clearer links between the criteria and risk assessments.

2 Part of the software design lifecycle involved observing IC's staff's working practices in order to identify work and data flows. It was found that whilst being observed the staff tended to follow the correct procedures set down by ICs. This did not necessarily happen whilst they were not being observed. This may have lead to a false analysis of the current domain. Hughes [Hughes94] stated that 'the reason why many systems fail is due to the fact that their design pays insufficient attention to the social context of work'. This application of ethnography involves the analysis of users' needs in relation to their working practices. The study of this relationship between ethnography and software engineering is still in its infancy. It is felt however that the inclusion of ethnographic studies as part of the software design lifecycle would have been beneficial and would have warranted further research.

10 Appendix

There should be way-points set within the cycles to say where fundamental changes can be made up to, and where smaller changes can be made up to etc.

2. Have more focused company input. Designate a small number of key users representing each of the system shareholders attributed to the requirements capture process. Those people should be knowledgeable of technology use and the key business processes.
3. Begin testing earlier in the cycle to catch the more fundamental errors more quickly and prevent unnecessary rework.
4. Use Boehm's spiral model of development. Set way-points within the cycle to formalise the development parameters; where stages commence and where they are completed.
5. Ensure that metrics are not used for political reasons. If one company manager sees the time it is being taken to develop another manager's system, they may argue why their own systems could not all be implemented.
6. Offer guidance with the analysis procedure. It is perceptually better to advise and the organisation, especially for SME's like IC, as to what can be realistically achieved in the way of a system, rather than to ask them what they want, and be given too many unworkable ideas.
7. Continue Process Refinement. Use the formalised process as a working document. It will not be correct the first time. Even the formal process should have a CP relating to it.
8. Time management. Support time management with the use of a tool such as Microsoft Project. This has the added benefit of not only enabling the company management to see how well the development is proceeding but also to see how their decisions impact the projects duration.

It was found that one of the key factors within the aspect of decision making was clear evidence that the more decisions that needed to be made, the more complex the management of the project. If it is possible to remove some of these decisions or to limit the people involved, the result would be more progress.

8.5 The creation of a Structured Business Method of Software Engineering

Creating a structured method or process is widely seen as a catalyst improving the way in which an organisation operates. Hammer [Hammer01] stated that with regard to process improvement, "The redesign of a company's organisation can achieve a quantum leap in performance." Although he struggles to prove the quantum leap, there is more than strong evidence to suggest that improvements can be made within a company's organisation by addressing its methods of operation and putting a more structured method in place. In an earlier article [Hammer90], he also stated "By re-engineering, we should obliterate them (the existing methods) and start over." Whilst obliterating a company's current operational processes may seem a radical approach, it would be worth consideration during a major business re-structure. Process improvement may be applied to software engineering processes as well as business processes.

ISO 9000 incorporated software process improvement (SPI) and encourage organisations to actively include software process improvement within their software development process. Since then, the European Union has funded over 450 process improvement experiments through its European Systems Software Initiative [Conradi02]. Rifkin [Rifkin02] claims that initiatives such as these have had only limited success. It is arguable that only limited success is better than no success at all. Indeed, other researchers claim as much as a 7 to 1 reduction in costs after implementing software process improvement [Ward01].

As part of IC process improvement, they had over a period of time made a concerted effort in building up a library of CP's. It had been requested that after the completion of the new system, a new CP be created for the SE process. Going on from that, it was decided to extend that concept to include a template for future developments.

This meant that there would be two documents:

1 A company procedure

2 A software development template (SDT). This would be used as a working document on which new developments could be based detailing the plan for implementation.

8.5.1 The Company Procedure

The CP would follow the format of current IC CP's detailing the expected standards and methods which should be promoted within the organisation. More specifically, it would highlight key areas within the process and detail how they should be tackled. This document should be phrased in such a way that it would be understandable to all company employees.

8.5.2 The Software Development Template

The SDT would allow IC to develop new systems either in-house or externally and would reduce the risks involved. It highlights key points within the lifecycle which are felt may need specific attention. By highlighting these points, giving examples and mandating specific information to be recorded, it allowed the SDT to act as a map for each particular project.

The SDT included advice on the following:

1 The collection of all relevant documentation relating to the process(es).

2 The creation of entities and any relationships in order to illustrate the current system.

3 Illustrating process flows.

4 Creating boundaries associate with the proposed system and project.

5 Formally stating what is required of the system and identifying functionality critical to the systems success.

6 Identifying stages within the project.

7 Setting target dates for those stages.

8 Implementing a CM procedure which would include relevant documentation

9 Guidelines for the development and testing cycles. This would be of use for in house development only.

10 The recording of details of point 9 such as start date, completion date, slip, and persons involved etc.

11 A review procedure applied to the entire process.

Figure 8.5 Points covered in the Template for Structuring a BPR Project.

The production of the template was an opportunity to ensure that the process would be as workable as possible whilst still offering the improvements which come with process improvement. IC are planning a trial of the template in the near future on two new systems; a contract management system and a job costing module.

Copies of both the new CP and the software development template can be found in the Appendix B and C respectively.

9 Conclusions

This chapter summarises the work and results attained in this thesis. The thesis examined the effects of introducing the creation of two functionally similar systems into an SME. The literature survey reviewed the current state of the art of Software Process Improvement and introduced a number of tools and methods which improvement can be gained. The conclusions are drawn focusing on the criteria for success as listed in Chapter 1.3.

9.1 Work Undertaken

The thesis undertook a study into an SME that had identified a problem domain within one of its current business processes. Previously, the SME had found the introduction of new software less than satisfactory and were concerned that the introduction of another new software system would result similarly. It was suggested that as part of the software development and implementation process, an evaluation be made of their current method of software engineering with a view to making improvements.

Part of the thesis was a literature survey, which aimed at highlighting software process improvement concepts, tools and methods. This literature survey was useful later in the thesis in identifying where IC could make improvements to their software design lifecycle. As part of this evaluation process, metrics were collected relating to the software design lifecycle and these too would be considered later.

A case study was prepared based on IC current position within the problem domain. This was useful in identifying the level to which IC understood the software design lifecycle as well as identifying the where the problems lay. The case study looked into the options available to IC and after no suitable COTS software could be found, it was suggested that a bespoke system be created.

At that point, the concept of the experiment was introduced; two functionally similar systems were to be created in parallel, but on differing computer platforms. One would be created in Windows and the second using Internet technologies. This gave the opportunity of comparing two sets of collected metrics and as such offered a unique situation.

After the completion of the two systems, one was chosen and used by ICs. The software design lifecycle was reviewed and a number of process improvements suggested to ICs. The thesis then went on to review the collected metrics and to identify a number of findings.

9.2 A Review of the Criteria for Success

The following sections review the criteria for success that were defined in Chapter 1.

9.2.1 To Demonstrate the Successful Application of BPR against a Real-life Scenario in order to create a Computer System to correct a Problem Domain.

The remodelling of the problematic business process was regarded by the company as a complete success. It is arguable that the implementation of a manual system could have achieved the same results; although this would have relied on the sales staff to police themselves. The computerised system also offered more than a manual system could in the form of management reports and control.
One of the most confounding facts of the whole project was to find that the implementation of the Windows system was regarded as such a success by the company when compared with the ratings as stated in Table 8.6. In those figures the Windows based system was given only given an average of 50% rating. If we refer to chapter 7.1.3.1, we find that the company management stated that they would only class a system as a success if at least 80% of the core requirements were achieved. The system achieved all the required functionality in Table 2.1 and therefore exceeded the 80% of core objectives. From this, it is possible to only assume that the ratings given in the questionnaires were inaccurate.

Referring to the figures in Table 8.6, it is possible to establish a number of facts based these figures. When considering these figures, one must make allowance for the threats to their validity; for example the development times may not wholly take into account the fact that one system was developed in as familiar programming language whilst the other was not. These are however realistic with regard to the constraints of IC. With this in mind, the following facts can be established:

1 It took 44% longer to produce the Internet based system than it did the Windows system.

2 It took 4 times longer to produce the GUI for the Internet based system than the Windows system. This (in part) was due in part to the 'false-start' of creating the code in CGI/C++. Even if this is excluded, it still took 60% longer to create the Internet based system.

3 Writing the code for the functions took 30% longer in PHP compared with Windows system.

4 27% of the time taken for the Internet Based system was spent on Researching and installing new software and hardware. This compared with only 7% for the Windows system.

The metrics showed that the creation of a system using a technology which was new to the organisation (the Internet) took the most time; approximately 44% longer. According to the questionnaire which the results of can be seen in Table 7.1, it was found that the Internet based system was also preferred by the users. Despite this fact we saw the company management opt for the Windows version. The outcome of the suitability of this decision will be clarified when the users adopt the system within their working practices. It was clear that the management made the decision not on the basis of functionality, but rather on NFR's. This is a clear example of NFR's having an overriding impact on system design and hence a clear motivation to ensure that attention is given to their collection and tracking during development.

McCall [McCall77] stated that "metrics can be objective or subjective". Although metrics can be interpreted from differing points of view and need to be appropriately applied to a scenario, in this case, the collection of metrics did prove a number of points:

1 Metrics are particularly successful at illustrating particular issues such as schedule slippage, the effect of additional requirements, time taken for coding etc.

2 Metrics are less useful in the use of measuring entities in which the users have difficulty in understanding the concept. An example of this is where the metrics collected from the questionnaires showed the Internet based system to be preferred, but the Windows system was the one to be chosen. This problem can partly be overcome by the use of Fuzzy Metrics [Ebert96] which consider the issue of uncertainty. Ultimately, the assessments are still relying on the user being able to understand a concept possibly too abstract for him/her.

Berard [Berard06] stated that “Although multiple metrics must be gathered, the most useful set of metrics for a given person, process, or product may not be known ahead of time.” This highlights the difficulty in identifying the correct metrics to collect. He goes on to state that “When we first begin to study some aspect of software engineering, or a specific software project, we will probably have to use a large number of different metrics“. This research identifies that the metrics collected were in some cases somewhat inaccurate; wrong metric being collected, not gathered accurately enough or misinterpreted. Much of the inaccuracy results from the subjective nature of the data gathering process for instance, the time taken to research and install the hardware and software relating to the Internet based version was high, but could have been classed as external to the project in as much as it was a one-off exercise. The inclusion of the metric inflated the total time taken for the Internet based system significantly. This point must lead to the consideration that the practice in the collection of metrics has the potential to become more accurate over time as experience of data collection is gained.

8.4 Outcome

The analysis showed that the decision making process needed to be changed and formalised for subsequent projects:

1. Formalise the CM procedure. Set CM parameters at points in the development cycle. For example, it is unacceptable requesting that an offline capability be implemented anywhere other that at the early design stage of the development.

Metric	**Measurement**	**Internet Based System**	**Windows System**
Software/ Hardware Research	Time	11	2
Software/ Hardware Installation	Time	22	4
Database Creation and implementation	Time	4	4
Lines of code	Lines of code	2670	2700
Initial Coding Time	Time	57	47
Testing	Time	7	8
Coding due to testing/ faults	Time	6	7
Additional Requirements	Quantity	4	4
Coding due to additional requirements	Time	13	11
Iteration Times	Time		
Iteration 1:		13	3
Iteration 2:		3	3
Iteration 5		4	4
Iteration 6		3	7
		34	26
		0	4
Total time taken	Time	120	83
User's system Grading	Mark out of 5. 5 being highest	4	3

Table 8.6 The Collected Metrics

It was decided that functional complexity was not a valid metric as both systems were functionally identical; therefore the metric would be indiscriminate.

B The Questionnaires

Ideal Caravan Sales Ltd
Transport Booking System Questionnaire
Management Version

Date completed:

Completed by:

Question	System A Web/Internet	System B Windows
How well do you feel the system has been received by staff?		
How would you rate the overall functionality of the system compared with your expectations?		
How reliable do you feel the system will be?		
What would be your overall score for the system?		

Please mark accordingly:
5 – highest
1 – lowest

Ideal Caravan Sales Ltd
Transport Booking System Questionnaire
Staff Version

Date completed:

Completed by:

Question	System A Web/Internet	System B Windows
How easy was it to learn the system within the given period of time?		
How easily could you negotiate around the system?		
How would you rate the data entry screen?		
How would you rate the Dairy screen?		
How would you rate the speed of the system?		
How would you rate the overall functionality of the system?		
What would be your overall rating for the system?		

Please mark accordingly:
5 – highest
1 – lowest

C The Company Procedure

57. Software Development Procedure

57.1 INTRODUCTION

The Development of Software whether in-house or externally involves a degree of risk. Ultimately, the question is; will the supplied System be what was required? This document has been produced with a view to creating a framework from which to base a new System Development around.

57.2 PURPOSE

By stipulating key areas within the Software Development process, risks can be minimised or at least recognised and controlled. The result of this procedure will be the production of a more suitable System.

The key areas are as follows:

- *Time Management*
- *The Development Lifecycle*
- *System Analysis and Design*
- *Staff Involvement and the Control of Input*
- *Testing procedures*
- *Process refinement*

57.3 THE PROCEDURE

57.3.1 Time Management

As with all projects, it is essential to keep track of time. We should be able to tell how the project is going in relation to set deadlines. To do this, we should use a time-management tool. One such tool is JXProject. This project management tool can be found on I:/Disk Copies/Time Management. It uses Gant charts to pictorially illustrate the current position within a project. Documentation is included.

Key stages (KS) within the development should be identified. Each KS should added to the Gant chart together with an expected completion date. We can then monitor time-slip.

57.3.2 The Development Lifecycle

A project can be controlled more readily by the introduction of a Development Lifecycle. The one most suitable to Ideal Caravans (IC) at this time is called the Spiral model. This method allows us to iteratively analyse, design, implement and test sections of the project in manageable chunks.
Before we can implement this, we must be able to identify the iterations. A good rule of thumb is to break the System down into areas, for example:

- *Graphical User Interface GUI*
- *Data entry*
- *Data output/displays*
- *Reports*
- *Security*
- *Functionality*

With this example, the first iteration wouldbe concerned only with the creation of a GUI. It may consist of:

- *Set time deadlines with JXProject*
- *Define Staff involvement*
- *Analysis and design of GUI*
- *Implemenation*
- *Testing*
- *Alterations*

The GUI would then be signed off and the next iteration covering data entry started. The skill needed in identifying these iterations may be reliant upon experience.

57.3.3 System Analysis and Design

The key to the production of a successful Computer System are;

- *The correct analysis of the current system*
- *The correct definition of the new system*

If we fail to get those two processes correct, the system may fail.

Analysing the Current System. If the system is being created by External Contractor (EC), it will be their responsibility to collect this information. Despite this fact, IC will be involved within the procedure.

- *Gather all relevant documentation*
- *Document all process flows*
- *Document data entities*
- *Document boundaries such as*
    - *Staff responsibilities*
    - *Users*
    - *System availability*
    - *Security*
    - *Environments*

Defining the Requirements for the New System. We must clearly define the Requirements of the new system. Even if an EC is responsible for this phase, IC must ensure that the Requirements are correct and will provide a complete System.

- *Define the existing System Specification*
- *Define the New System's essential Specifications*
- *Define the New System's desirable Specifications*

In order to correctly identify Requirements, we should identify those aspects which are critical to the success of the System. If we can manage to quantify those aspects, we will be able to ascertain the level to which those aspects have been achieved. These are called Critical Success Metrics (CSM's). As an example, consider if Ideal Caravans wished to introduce a simple System to track a van's profits. The van may be bought and sold a number of times over its life time. We could identify CSM's such as:

1. *The ability to identify the profitability of a van. How expansive is the functionality relating to this function? Does it take into account external factors such as; Interest rates, Buy-in condition, repairs/renewals, transport costs etc.*
2. *How user friendly is the displaying of this information; Is it text only, charts, numeric etc. Can we measure the ease of identifying traits. How critical is this?*

When you consider the Requirements of a System, attempt to use this method or at least consider it.

57.3.4 Staff Involvement and the Control of Input

The more Staff which are involved in any process increases its complexity; more ideas, more conflicting interests, more opinion. To control this, identify key personnel involved in each process and involve only them. Try not to allow input from other Users who are not Key Personnel.

57.3.6 Testing Procedures

Testing should take place at each iteration cycle. In our example above, the GUI was tested and any amendments made before moving onto the next cycle. Any major changes identified may need to be documented and addressed later. Minor changes may be added ad-hoc depending upon their size, and the time constraints.
Involve only the Key User(s) in this process, and attempt to isolate only the process being tested. This will help to focus the mind of the 'Tester'.

57.3.7 Process Refinement

At the end of the project, the processes used during the Software Engineering should be examined. If improvement can be made, implement the change and document them together with the reasons.

57.4 OTHER INFORMATION

This document should be used in conjunction with 57.4a which is a template for the Software Engineering process.

D The Software Development Template

Ideal Caravans Software Development Process

1 Introduction

This document is design to be used alongside the development of a new Computer System. It is a working document and needs to be kept up to date on a daily basis. You will need to make copies of certain pages as and when they are needed.

This document is not necessarily in chronological order. The preferred method of software design is using Boehm's Spiral model which allows the reiteration of certain aspects of the development process.

It is recommended that the document is used in conjunction with some time management software such as Microsoft's Project or JxProject.

2 Analysis

2.1 Analysis of Current system:

2.1.1 Documentation

Gather all relevant documentation. This process is fundamental to a successful analysis of the problem domain. This is the 'what'. What we are collecting, inputting outputting, printing, storing etc. Any piece of paper which is relevant to the current process which you are trying to remodel should be collected and documented.

Examples of such paper would be the following:
Invoices, chits, delivery notes, timesheets, diaries, lists of products, receipts, emails, print-out etc.

When all of the paper work is collected, we will end up with a summary of the data which the system is managing. We can then use a formalised method such as an Entity Relationship Diagram to illustrate the relationships between the data.

Place All Gathered Documentation Here

List Entities and Place them Here

Create Entities Relationship Diagrams and Place them Here

2.1.2 Process flows

Having collected the 'what', we need to collect the 'how'. This is the process flow, the method by which the data is gathered, input, stored, retrieved and used.
To illustrate this, we may need to use a formal method such as the Unified Modeling Language. Data flow is more easily visualised by use of pictorial diagrams.

For example;
We may need to illustrate how an order comes into the Company from outside, it may come in by numerous means; email, telephone, letter, word of mouth etc. We need to know how it comes in, who receives it, how it is processed, where it is stored etc.

In many cases, when a process is automated, it may need to be redesigned as part of the Design stage. We do not need to consider this here.

The key point here is that every single Entity listed previously should appear in a data flow. If it cannot be fitted in, reconsider its existence.

Create Process Flows and Place them Here

2.1.3 System Boundaries

System boundaries could be anything which have a metric associated with them. We need to do this in an analytical manner.

Primary boundaries are such items as start time, duration and priority.
Secondary boundaries are items such as;

- Staff/Management involvement. Who is to be involved in any of the processes? What are the guide lines to their involvement?
- Who are the target Users? What is their P.C. literacy level?
- Security levels. How secure does the data need to be?
- What are the physical boundaries of the system? Single P.C, networked, Internet etc.

These should all be documented at this stage.

Place System Boundaries Here

3 Design

3.1 Specification of Requirements

Here we need to specify in writing what the new system should achieve. This is concerned with the data which needs to be stored and the functionality attached to that data.

This entails documenting the

Current system entities.
Current system functionality
Any new entities
Any new functionality

This process also involves the prioritising of those functions which are absolutely necessary and those which could be regarded as a luxury.

List the New System Entities Here

List the New System Functionalities Here

3.2 The Change Management Procedure.

The management of change within an implementation of a new system is very important. It is possible to allow a project to become completely blown off course by late addition to the Requirements.

There should be a cut-off point in the form of dates which specifies when the latest alteration may be made to a stage. After that point, any proposed changes should be categorised into importance and documented for implementation into later system releases.

Place Change Management Documents Here

Change Management Document

Number: .....

Date: .... **Time: ....** **Requested By: ....**

Delete as appropriate: Addition/Alteration/Removal

Reference: **...................**

Description:

Schedule for review (date): ....

Priority: **........**

4 Development and Testing

The spiral model lends itself to a prototyping method of development. The functionality of the system should be broken down into key stages, sub stages and further stages as necessary.

Start development from a primary key stage for the system. In most cases, this will be the GUI, but could also be Reporting functions or the Database. Implement the stage and allow the chosen Company representative to test it. Reiterate this process until the stage is completed as time allows.

Each stage should be documented in the Gant chart for time-management purposes. The dates should correspond to those in the table below. This is a way of cross checking that all requirements will be attended to.

Each stage should be developed in isolation as much as is practicable.

Record of Development

Stage	Sub Stage	Requirement	Start	Tester	End	Spare

5 Project Analysis and Recommendations

On completion of the project, the team involved should have a final meeting and make recommendations for future projects. Any changes should be documented and the alterations made to the Company Procedure and to this support document. Only by re-evaluating and refining this process will it succeed in becoming a Company Procedure.