A CASE STUDY OF THE WEB-BASED INFORMATION SYSTEMS DEVELOPMENT.

John Paynter and Michael Pearson

Department of Management Science and Information Systems, University of Auckland, Private Bag 92019 Auckland, New Zealand

J.Paynter@auckland.ac.nz, Michael.Pearson@rocketmail.com

The WWW is a technologically active environment, and information systems developed for the WWW differ in significant areas from traditional applications. Existing literature has suggested that software development methodologies for traditional applications may have to be modified, or indeed replaced to meet the needs of the WWW. Consequently, a number of software development methodologies have emerged to support the development of WWW-based information systems.

This research explores the current software development methodologies used by organisations in developing WWW-based information systems. A case study approach is used to investigate, in the context of the organisation, how various WWW-based information systems are developed and the reasons why particular strategies are used. The organisations were selected on differences in type, size and information systems developed. The cases were analysed based on the software process model, methodology, tools and techniques within an organisational context.

The core finding of the research was that the development of WWW-based information systems is dominated by the challenges presented by new technology. In addition, organisations take a structured problem solving approach rather than adopting methodologies specifically designed for the WWW. The results of the research also indicated that deficiencies existed in the development strategies used, principally in the area of inadequate guidelines and lack of documentation.

1. Introduction

Over the last three years the focus of the information technology industry has moved towards development for the World Wide Web (WWW). Information systems using WWW technology, delivered by an Intranet or via the Internet, are now prevalent throughout New Zealand and overseas.

Within New Zealand, a wide variety of organisations are deploying information systems onto the WWW, including banks, government departments and other service providers. They are using the WWW as a strategic business tool, supporting their existing operations or providing a low-cost solution for delivering a new product or service line.

1.1 Proposition and Aims

There is an abundance of information on the graphical and user interface aspects of WWW site design. In addition, a large body of knowledge has been developed in the area of software development methodologies. However, there has been very little research conducted to examine if these existing research methodologies are applicable to the information systems developed for the WWW. The WWW is a technologically dynamic environment, and presents new challenges for developers. In order to address the emerging WWW environment, a number of researchers have proposed software development methodologies specifically for the WWW.

Consequently, the aim of this research was to investigate how organisations are currently developing WWW-based information systems, and the reasons why those methods are being used. This examination is important because, as the literature demonstrates, one would expect to observe changes in the software development methodologies used by organisations developing software for the WWW. This study also provides a valuable contribution, by addressing the absence of detailed evidence concerning whether software development methodologies are keeping pace with technology.

In the following sections an examination of the WWW is given, and a discussion provided of specific methodologies that may prove useful for developing information for the WWW. Section 3 outlines the research methodology, and discusses the need for examining the development of an information system in the context of the organisation. The specific research questions are detailed, supporting the main aims of this paper.

The case histories of three organisations are then given, followed by an analysis of the case studies based on the research questions. Within this analysis, the deficiencies that exist in the development of WWW-based information systems are outlined, along with details of how the development for the WWW differed from other classical systems.

The central finding of this research was that the development of WWW-based information systems is dominated by the challenges presented by new technology. Organisations adopt conventional methods, and apply them to the unfamiliar environment of the WWW.

2. The World Wide Web

Since its inception, the WWW has come to stand for a number of different concepts (Berners-Lee, Cailliau, Luotonen, Nielsen & Secret, 1994). The WWW incorporates the idea of a boundless information world in which all the items have a reference by which they can be retrieved.

The late 1990's has seen the WWW emerge as a strategic business tool, driven by the commercial interests of companies. The original vision of Tim Berners-Lee of the WWW being an open-architecture is now under threat, as competing technologies challenge each other for market dominance.

Organisations who are developing information systems based on the WWW now find themselves in a difficult position. In an attempt to gain a competitive and strategic advantage via the WWW, they must follow the latest technology. The organisation's WWW presence must have a high level of technical sophistication and functionality, and an outstanding design if they are to benefit from using the WWW (Cronin, 1996). Furthermore, there is the challenge of shifting and integrating existing business practices to take advantage of the interactive opportunities presented by the WWW. A strategy is required to manage the development of these WWW-based information systems to accommodate these new technologies in the present and in the future.

2.1 Intranet and Extranet-based Information Systems

In an attempt to take greater advantage of WWW and Internet technologies, organisations have recently begun to segment the concept of the Internet into subconcepts called Intranets and Extranets. The terms Intranet and Extranet reflect differing levels of managed access to internal Internet and WWW-based information systems.

Telleen (1997) has identified three sources of information that emerge on an organisation's Intranet. The first is formal information, and is the officially sanctioned and commissioned information of the organisation. Project or group information is also provided and is intended for use within a specific group, and finally informal information also appears within the Intranet when authors and users discover the ease of publishing information and sharing ideas.

What we are now seeing within organisations is the adoption of the Intranet and Extranet concepts as a means to disseminate more complex information. Telleen's three categories of information are being extended by the rapid development of WWW-based information systems within organisations as they attempt to gain a strategic advantage over their competitors .

2.2 WWW-based Information Systems

The applications that have emerged on Intranets and Extranets using WWW technologies, can be referred to as WWW-based information systems. However, a WWW-based information system should be distinguished from a standard WWW application or page; the nature and the type of information that is made available to the user is different. The standard WWW page is uni-directional in the way it provides information to the user; with the user often making requests in catalogue or directory based sites by following hypertext links. A WWW-based information system can be defined as an application that not only disseminates information, but also proactively interacts with the user to aid them in their task (Takahashi & Liang, 1997). Information is therefore presented to the user in a bi-directional manner in a WWW-based information system.

While this definition of a WWW-based information system is generalised, there are commonalties between such systems. They adopt extensions to the basic HTML, using Java, Java Script, or VB Script to provide additional functionality to the basic hypertext links. Dynamic creation of HTML pages are also common in WWW-based information systems, using a connection to a relational or object-oriented database to process user requests. This is becoming increasingly more important in the development of electronic commerce environments, such as on-line shopping, where there is a need to process user requests while also keeping the information on the WWW page up to date (Yamamoto, Kurokawa, Tokumaru & Adachi, 1996). The general structure of a WWW-based information system, as can be seen from Figure 1, follows the architecture of most Intranet and Extranet applications.

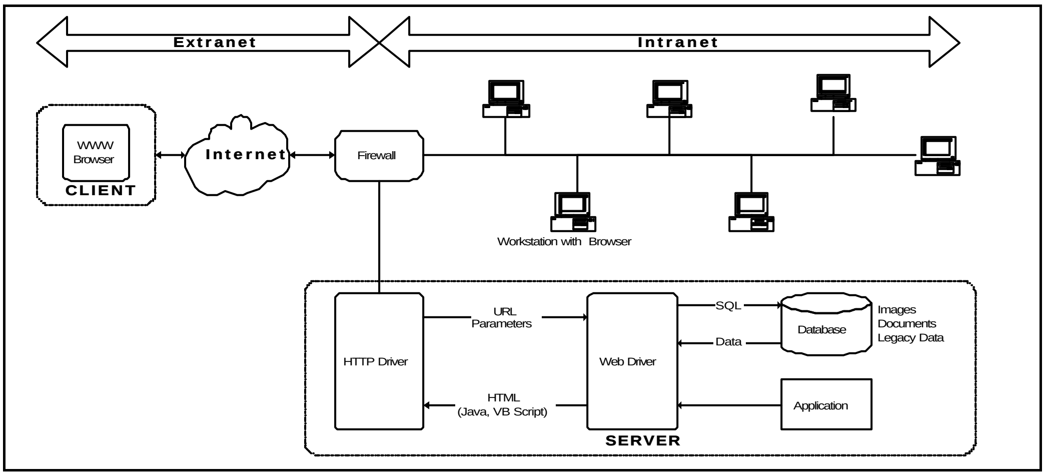

Figure 1 An example of an Intranet and Extranet implementation. (Bouman, 1997)

The emergence of database centric applications on the WWW has led to a change in the nature of WWW pages. This emerging change is supported by the observations of Gellersen, Wicke and Gaedke (1997), as they categorise WWW pages as being either static or dynamic. Static hypertext systems, as commonly provided by standard WWW pages, can be characterised by both static links and static pages. Database-centric applications are characterised by dynamic page creation but have a static link structure. Dynamic applications, however, provide both dynamic link structure and page creation. WWW-based information systems fall into the database-centric and dynamic application classes.

A WWW-based information system therefore provides structured as well as unstructured information as described by Bichler & Nusser (1996a, 1996b). In a similar manner to Gellersen et al., they consider database-centric WWW-based information systems as providing highly structured information with high volatility. On the other hand, dynamic WWW-based information systems that provide more multimedia services, support unstructured information with low volatility.

WWW-based information systems are fundamentally different from traditional systems in several critical areas. Bichler and Nusser (1996c) and Bieber and Isakowitz (1995) suggest that WWW applications often involve people with differing skill sets, including authors, content designers, artists as well as programmers. WWW applications also involve the capturing and organising of the structure of a complex information domain, whilst making it clear and accessible to the user. Yourdon (1996) suggests that the difference that exists between older technologies and WWW applications means that the 'older' methodologies and techniques, as given in Table 1, need to be modified, and in some cases replaced entirely. As a result, Yourdon proposed that a focus should be placed on three key areas:

?? the development life cycle.
?? methods and techniques for the life cycle.
?? criteria for choosing Internet (WWW) technologies.

However, the current focus within the WWW community is on technology. It is technology that is the driving force in how the WWW is evolving (December, 1996; Morrel, 1997). This has resulted in a lack of guidelines for a structured design process that will meet the requirements of these new technologies.

2.3 The Emergence of WWW Development Methodologies

The arrival of new Internet and WWW technologies has in many ways seen history repeating itself. Yourdon (1996) draws parallels from the client server revolution that provided strong arguments for replacing the sequential waterfall life cycle model with an iterative rapid applications development (RAD) life cycle. WWW applications are likely to accelerate the RAD process to the point where it becomes 'FAD', or Frantic Application Development. The technological driven nature of the WWW is forcing developers to deliver applications that use these new technologies, such as streaming video and audio, in shorter and shorter time spans. This is spurred further by the development efforts of vendors, including Netscape and Microsoft, who continually release new versions of their WWW browser software. These pressures from technology change have the effect of encouraging ad-hoc development methodologies to be used in an attempt to meet demand (Yourdon, 1996; Gellersen et al., 1997).

The dramatic growth of the WWW is now leading to larger scale applications, distributed over a number of sites. Such WWW-based information systems are no longer manageable with ad-hoc methods. This, combined with the rapid technological change and the differences that WWW-based information systems present from traditional systems, has led to a need for new software development methodologies, techniques and tools. Bieber & Isakowitz (1995) suggest that hypermedia applications, which are at the core of WWW-based information systems, involve many different components such as navigation, user interface and control storage. As a consequence, data models such as E-R diagrams, data flow diagrams and object-oriented hierarchies cannot represent the design interfaces they encompass. Furthermore, such hypermedia systems are becoming increasingly hard to maintain due to a lack of a structured design process.

Conventional software development methodologies are well established and provide guidelines for managing complex tasks in the development of traditional systems. But, as has already been alluded to, WWW-based information systems differ significantly from traditional applications, which are the focus of the established methodologies. Certain elements from these methodologies may be adopted for the development of WWW-based information systems, but multimedia aspects of the WWW raise numerous difficulties (Bichler & Nusser, 1996a, 1996b). It is for this reason that Isakowitz, Stohr, & Balasubramanian (1995) emphasise that hypermedia design, and subsequently the design of WWW-based information systems, is "more of an art form than a science".

There currently exists a large body of literature on the graphical and user interface aspects of WWW site design (Nielsen & Sano, 1995). User interface design is an important aspect when creating a WWW-based information system, due to the visual appeal of the WWW. Nielsen and Sano's study, using discount usability engineering, supports this view by showing that a uniform user interface structure can make a WWW-based information system significantly easier to use.

However, Takahashi & Liang (1997) present the argument that such approaches only emphasise the visual design of a WWW page, but do not provide a systematic way of designing the structure of WWW sites and applications as a whole.

A number of methodologies have emerged to address the need for structured development guidelines for developing WWW applications as a whole. These WWW specific methodologies have their origins in hypermedia and hypertext methodologies.

Hypermedia design is closely related to the field of database design, with many hypermedia methodologies using database design techniques. These models, used during the design phases, help structure the information and lend to increased clarity and easy information retrieval within a hypermedia application. By specifying the hypertext structures, developers can avoid structural inconsistencies and mistakes, such as missing hypertext links. The traditional hypertext and hypermedia methodologies include Hypertext Design Model (HDM) (Garzotto, Paolini & Schwabe, 1995), Object Oriented Hypermedia Design Model (OOHDM) (Schwabe & Rossi, 1995) and Relationship Management Methodology (RMM) (Isakowitz, Stohr & Balasubramanian, 1995).

When a WWW application has a dynamic structure and highly volatile data, then the hypermedia models (HDM, OOHDM) and the RMM methodology can offer little assistance, as they focus on structured information domains. In the case of a WWW-based information system that has a dynamic structure, it may be possible to partially address development issues via RMM by separating out the structured portion of the information domain (Isakowitz et al., 1995). What is needed are specific software development methodologies for the WWW that address the dynamic and unstructured requirements of WWW-based information systems.

In the area of software development methodologies for the WWW, there is a limited amount of formalised literature. However, a number of methodologies, techniques and tools are beginning to emerge that directly address this issue. These methodologies include the WWW Design Technique (W3DT), also known as the Structured Hypermedia Design Technique (SHDT) (Bichler & Nusser, 1996a, 1996b, 1996c), a WWW development methodology proposed by December (1996), and a methodology proposed by Takahashi and Liang (1997).

2.4 Research Opportunities

The WWW is a rapidly changing environment driven by technological advancements, and the perceived need by organisations to gain competitive advantage through a WWW presence.

For methodologies, tools and techniques to evolve in such an environment, Wynekoop and Russo (1997) have proposed that research to assess the current systems development practices within organisations would be expected. Furthermore, one would expect to see fundamental changes taking place in software development methodologies within organisations operating in a technologically dynamic environment, such as those developing WWW-based information systems.

However, there is no empirical evidence that software development methodologies are keeping pace with technology or organisational change (Wynekoop & Russo, 1997), hence a similar argument may also apply to the WWW.

There has been little evaluation of software development methodologies in use, or in the selection, adoption, development or use of methodologies in practice. Much information systems (IS) development research implicitly assumes that software development methodologies are used and are effective, but such claims have not been validated (Wynekoop & Russo, 1995).

Due to the lack of research in software development methodologies, it cannot be concluded that existing methodologies are either appropriate or inappropriate for future systems. In the context of the WWW, this means we are unable to evaluate the effectiveness of existing or proposed methodologies.

To address this shortcoming concerning the WWW, an understanding must be gained in how WWW-based information systems are developed, by evaluating the currently used and proposed methodologies and finding out what is needed before suggesting new advances in methodologies, tools and techniques.

Furthermore, a clear understanding of the present development environment is necessary to guide the evolution of methodologies for the future. This view is supported by Westrup (1995) and Wynekoop and Russo (1995, 1997), who have suggested that a fruitful area of research, that will provide insights into the effectiveness of software development methodologies, will involve observation of the practice of software developers and the study of methodologies in use.

By taking such an in-depth approach, insights can be gained on how WWW-based information systems are currently being developed, and why the particular software development methodology is being used, in the context of the participant's perspectives.

3. Research Methodology

The objective of this paper is to explore the contemporary phenomenon of WWW-based information systems. The software development strategy, specifically involving analysis and design methods, used by organisations in developing WWW-based information systems will be investigated. The study is exploratory in nature as the analysis and design of WWW-based information systems does not enjoy an established theoretical base.

Software development methodologies such as the Waterfall Model (Boehm, 1976, as cited in Comer, 1991), Spiral Model (Boehm, 1988) and prototyping have a strong theoretical base.

Furthermore, a number of methodologies have been suggested to aid in the development of hypertext and multimedia applications, and these may have a place in WWW-based information systems development (Garzotto, Paolini & Schwabe, 1993; Isakowitz, Stohr, & Balasubramanian, 1995; Schwabe & Rossi, 1995).

However, there has been very little, if any, research conducted on how these analysis and design methods can be applied to WWW-based information systems.

3.1 Case Study Research Design

The case methodology is useful when contemporary events are the focus of research, and the phenomenon is not supported by a strong theoretical base. This approach also allows the researcher to answer "how" and "why" questions in an area where few previous studies have taken place (Benbasat et al., 1987; Yin 1994).

The case methodology is therefore applicable for the following reasons:

?? The study of the design of WWW-based information systems cannot be easily studied outside its natural development environment.

?? The research questions directly address a contemporary event; the analysis and design of information systems for the WWW.

?? WWW-based information systems, and their design does not enjoy an established theoretical base.

There are however a number of criticisms of the case study as a qualitative method. There is an inability to control and manipulate the independent variables and a risk of improper interpretation (Gable, 1994). These problems can be overcome through careful research design (Lee, 1989; Yin, 1994).

Within information system's case research strategies, a number of approaches are available (Benbasat et al., 1987; Lee, 1989; Yin,1994). In this research, the five guidelines proposed by Yin (1994) will be used: the study's questions, its propositions, the units of analysis, the logic linking the data to the propositions and the criteria for interpreting the findings.

3.1.1 Case Study Questions

The overriding question of this research is "How and why do developers use specific software development methodologies for WWW-based information systems?".

The specific case study questions are based on a methodology research taxonomy proposed by Wynekoop and Russo (1997). They investigated the research methods used in studying system development methodologies and outlined a number of questions that must be addressed. These questions included the use, selection, development, adaptation and evaluation of systems development methodologies.

This research will therefore focus on the following questions:

?? How is the development of information systems for the WWW presently realised?

?? How and why is WWW-based information system development different from classical IS development?

?? What deficiencies can be identified in the development of WWW-based information systems and how can they be overcome?

3.1.2 Propositions

The case study questions are exploratory in nature, due to the lack of knowledge in the field of WWW-based information systems development. This study therefore is not based on any pre-existing hypothesis.

However the rationale behind the study is based on the rapidly changing nature of the WWW, and the new technologies and challenges it presents. In this environment "one would expect to see fundamental changes taking place in systems development methodologies" (Wynekoop & Russo, 1997).

This suggests that new development methodologies may be required for WWW-based information systems.

3.1.3 Unit Of Analysis

The unit of analysis for the case study will be a WWW-based information system that has been developed within a particular organisation. This will involve a study of the systems development methodologies undertaken by the organisation in developing the WWW-based information system.

The organisations will be chosen for their 'theoretical' replication, that is, chosen such that contradictory results are expected (Yin, 1994). Through contrasting approaches, it is hoped that critical elements of the development process can be highlighted. This will aid in formulating theory for the development of WWW-based information systems.

3.1.4 Linking Data to Propositions

To explore the nature of WWW-based information systems development, data will be gathered from multiple sources.

These data will be used to provide insights into the software development methods used by organisations for WWW-based information systems.

3.1.5 Criteria for Interpreting Findings

The findings of the case study will be interpreted via a cross-case analysis. Commonalties and differences between the case sites will be used to further develop theory in the development of WWW-based information systems. Where possible the contextual and data richness of the study will be presented and a clear chain of evidence will be established (Benbasat et al., 1987).

3.1.6 Case Study Site Selection

A multiple-case design will be used, as the intent of the research is to explore how and why organisations perform the analysis and design of WWW-based information systems. This will allow for cross-case analysis and the extension of theory (Gable, 1994).

The objective for this research will be a "theoretical" replication, so the case study sites will be selected based upon differences in organisational size, culture and the type of information systems developed. All the organisations will have had experience in developing classical IS systems, so that the development methodologies used can be compared with those for the WWW-based information system.

It should be noted that six organisations were approached to participate in this research, after gaining approval from the Human Subjects Ethics Committee at The University of Auckland. All organisations were guaranteed complete confidentiality, and the prospective participants were given an opportunity to review interview transcripts and written findings. On two occasions the researcher offered to conduct a review of all relevant documentation on site, taking written notes rather than copies.

Three of the organisations withdrew expressing concerns regarding the nature of the data to be gathered. In one case, the organisation felt that the way they developed a WWW-based information system was commercially sensitive, and withdrew from the case study.

Another organisation, that had developed a WWW-based information system for a large New Zealand bank, withdrew due to an unwillingness to share their "commercially sensitive methodologies". This is despite an enthusiastic response from the bank's project manger. It was later discovered that the organisation in question had no documented strategy for developing the bank's WWW-based information system. The third organisation that was approached was a government department, and they declined as they were unable to justify the resources required.

4. Case Studies

4.1 Alpha Software Limited

Alpha Software Limited (ASL) is a software development company based in Auckland. It currently employs a staff of seventy across a range of six service divisions: Software Services, Engineering Services, Product Distribution, Multimedia And Creative Services, Training Solutions and Project Management.

The Software Services team contains twenty analyst/programmers who work in small client-focused teams to provide custom software solutions. Most of the staff are university educated and are experienced in aspects of software development, with specific areas of expertise involving: client/server development, systems integration, workgroup computing, executive information systems and contact management systems.

The company has been operating successfully in the New Zealand market since its inception in 1983, focusing on a narrow market segmentation involving predominately affiliates of multi-nationals and leading corporate clients. This has recently led to a number of opportunities in overseas markets.

During March 1996, one of ASL's clients, Gamma Business Services Limited (GBSL), approached them to create an Order Entry System, to augment the service offered to its key customers and to provide a competitive advantage from improved communication with these organisations. GBSL is a leading global supplier of document information, print outsourcing and database marketing, with 19,000 employees and sales in 1996 of US$2.5 billion.

One of GBSL's key clients, a leading New Zealand corporation, asked GBSL to provide an on-line system to enable them to order business cards and associated paper products. GBSL envisaged that such a system would create new business opportunities by providing its clients with an electronic platform for delivering and receiving information. The proposed system would also enhance GBSL's position as an integrated information service provider and global technology leader.

The choice of ASL for this case study was based on their experience with client/server technologies involving Visual Basic, Powerbuilder, Perl, Sybase and SQL Server. The Intranet-based Order Entry System was ASL's first attempt at developing an Internet or Intranet solution for a client.

ASL therefore provided an opportunity to explore how a company with previous experience with successful client/server implementations approached the development of a WWW-based information system.

The research was conducted in July, 1997, towards the end of the implementation. Over this period an historical reconstruction was made of the development methodologies, techniques and tools used during the entire project. The principal method of data collection was in-depth interviews with a range of ASL participants. In order for the participants to become more comfortable with the interviewer, a number of interviews with the same participant were conduct ed. In addition, a number of informal discussions were also held.

The personnel interviewed included the two developers, Chris and Aaron, and anecdotal evidence from Jeremy, the Sales Manager, and Rachel the Account Manager.

Chris and Aaron formed the basis of the Internet Solutions team within the Software Services group of ASL. A third member, Tony, was recruited from within Engineering Services. The Internet Solutions team was formed specifically for the development of the on-line system for GBSL, and followed the well established practice within ASL of small client-focused teams.

Both the developers had no prior experience with WWW technologies, apart from using the Netscape browser to view WWW sites. However, they are familiar with Visual Basic, C++, Perl and Sybase. Tony, a systems engineer, had specific skills in the area of TCP/IP connectivity and PC to UNIX integration, and was familiar with some WWW technologies through his role as administrator of ASL's WWW home page.

Rachel had a large degree of exposure to client/server systems in her role as analyst/programmer. Her current position as Account Manager for GBSL gave her a unique insight into their business requirements, and as a result had a strong professional relationship with Simon, GBSL's Information Technology Manager.

4.1.1 The Intranet-based System

The Intranet-based Order Entry System implemented by ASL consisted of three broad functional areas: order entry, contact management and ordering of business cards.

The system was implemented using a variety of WWW technologies. Perl was extensively used in the generation of product tables, along with JavaScript and Java applets for the validation of data and display of graphics. GBSL saw such an implementation as providing leverage over other competitors as it would allow centralised control of the software and, as a result, easier maintenance.

An implementation using the WWW also provided wider scope than a traditional client/server implementation, as the system was going to be at least New Zealand wide. The initial implementation was however Intranet-based, with a future stage being a move to the Internet.

4.1.2 Development Strategy

Initial meetings regarding the scope of the system occurred in early April 1996. Rachel, ASL's Account Manager, had been liaising with Simon, the principal contact and stakeholder for the project within GBSL in New Zealand. At this early stage, the development of a prototype was proposed that would be demonstrated to one of GBSL's clients, after which the development of the full Intranet-based Order Entry system would proceed.

During the latter part of April and early May 1996, Chris and Tony from ASL became involved with the creation of a prototype. They liaised with Rachel on its content, as she had a more rounded knowledge of GBSL's business requirements. This prototype was used as a pricing gauge for both ASL and GBSL, and included some of the functionality required by Simon in his discussions with Rachel. As Chris commented, the prototype was essentially a throwaway:

"…it [the prototype] was to prove it would work, but it didn't prove enough. We didn't realise what was involved with the new technologies we were dealing with. Looking back on it we should have done a complete "proof-of-concept". We did a prototype, but it didn't really connect to the AS/400."

It appears from this statement that the early stages of scoping the project lacked thought and planning in how the technologies would work and interact. Chris and Tony were both working on other projects at the time of the initial discussions. They became involved on the project together for about a week, at the request of Jeremy and Rachel, to provide a prototype to aid in the sale of the system to GBSL management and their clients.

Much of the prototype development involved learning how to create screens using a WWW browser, due to a lack of experience in such technology on the part of the developers. Tony had spent a significant amount of time researching the connection between GBSL's AS/400 and the Sybase database.

GBSL was initially proactive and enthusiastic, especially during the development of the prototype. However, after the prototype, some five months passed by without any commitment from GBSL, despite their initial desire to go to market with the system as soon as possible.

During this five month period, GBSL was preparing a formalised functional specification. It became apparent that GBSL did not want ASL to write the functional specification due to budgetary constraints. As previously alluded to, a prototyping philosophy was prevalent within the Software Services group at this time. It was unusual for ASL to allow its clients to develop any specification without direct input from its developers, due to problems that had arisen in previous projects. Furthermore, a perceived need to reduce documentation, and ultimately the cost of the project, by using a prototyping philosophy, led to GBSL being allowed to create a functional specification.

However, this approach proved to be inadequate as explained by Chris in the following quote:

"They [GBSL] wrote the specification, then I wrote my own from that as it was very vague. They didn't want to spend the money having us write it. Basically, I didn't have the opportunity to write a technical specification. I spent two weeks with Rachel "beefing up" the functional specification with technical information on how data was to be structured on the AS/400."

Chris also noted that the functional specification provided by GBSL was written by someone who had little technical exposure, but was realistic in what was required. Due to a lack of input from the ASL developers, numerous problems arose. For example, in the functional specification, statements about the existing AS/400 architecture were made which later proved to be false. Chris and Aaron both suggested that this was a result of a lack of understanding by GBSL of their core systems.

The functional specification was delivered by GBSL on August 8 1996. The specification had been created by Simon and reviewed internally by GBSL. The document provided an overview of the proposed system, split over three phases. It was envisaged that by phasing the project, the order entry system could be built up as a series of independent base modules that could be selected or deselected for inclusion on a client by client basis.

On October 11 1996, a formal request by Simon at GBSL was made to the ASL Sales Manager, Jeremy, for the development of the Intranet-based Order Entry System, based on their functional specification. An unrealistic expectation of two weeks was made by GBSL for the development and installation of the system.

During October 1996 through to late November 1996, the functional specification provided by GBSL was completely re-written by Chris and Rachel. This involved liaison among Chris, Rachel and Simon. The refinement process involved e-mail communication between the parties and revolved around the lack of technical detail in the functional specification.

4.1.3 Design Issues

The majority of the development occurred during November and December 1996, with final completion in February 1997. The design of the system was conducted concurrently with development.

The lack of experience of the developers using such technologies as Java and HTML was reflected in the approach taken in the design and development of the system and in the way it progressed. This produced much frustration within the development team as progress was very slow due to an abundance of technical challenges.

The development team experienced problems on three fronts: the Java development, pre-packaged software and security. The problems stemmed from a lack of developer experience and from bugs in the software being used.

Java was chosen because of its virtual machine paradigm; it was able to run on any hardware or software platform using the Java virtual machine or Java compliant WWW browser. However during development, Chris and Aaron found the system would operate differently between different versions of a WWW browser, even from the same vendor.

The technical difficulties experienced by the developer led to the first two phases of the development being combined. The first phase of the project was to have been tested in the field, but the technical problems forced a rapid design and implementation of the second phase involving dynamic business card orders.

The general design philosophy had been to structure the development for re-use. However, the technical challenges that became apparent meant that much of the focus shifted towards getting the project completed by ensuring the functionality was available to the user. Chris was quoted as saying:

"It had to be structured well, but we had to get things working."

4.1.4 Present Situation

The Intranet-based Order Entry System was installed during July 1997. It was 200% over budget and continued to experience technical difficulties. Additional code was required to accommodate the variety of WWW browsers used by GBSL's clients.

GBSL have been enthusiastic about the system, and have used a graphic design company to enhance its WWW home page. At this stage, there are no plans to move the system over to the Internet. The current architecture developed by ASL will not support security across the Internet. Chris and Aaron have suggested that the system will need to be completely rewritten to accommodate the additional security concerns, possibly using a combination of Active Server Pages and Microsoft Transaction Server.

In reviewing the project, Chris and Aaron would have preferred using an integrated Microsoft solution running Windows NT, with Internet Explorer for the client WWW browser. In their view this option would have removed many of the technological barriers.

Chris summarised the project as follows:

"…people didn't know what they were doing etc. It would have to have been one of the worst I have worked on though for that. I just wanted to get the thing running."

Jeremy and Aaron later commented on the high risk factor within the project, as

After the completion of the project, ASL management decided to disband the Internet Solutions team. The main reason given was the large number of technical problems experienced during the development of the Intranet-based Order Entry System, and also the rapidly moving technology advancement on the WWW.

4.2 Beta Communications Limited

Beta Communications Limited (BCL) is based in Auckland, and specialises in providing business solutions to a range of domestic and international organisations. BCL currently employs a staff of eighteen across a wide variety of skills and services.

BCL was established in 1995 to provide specialised development and delivery of Internet and related communications technology solutions. Prior to 1995, BCL formed part of a larger organisation that specialised in software development. In 1995, this organisation separated to form two organisations. The software development component was absorbed by a multi-national management consultancy to establish their Information Technology division in New Zealand. The remaining departments, consisting of communications technology specialists and network engineers, formed BCL.

The services offered by BCL include providing project management, integration of package solutions, network design, integration and tuning, and the delivery of Internet, Extranet and Intranet solutions to domestic and international clients.

BCL has provided consulting services to a number of international airlines and domestic telecommunications companies. In the past two years, BCL has been involved in the development of seventy WWW projects, including the largest commercial site in New Zealand, which focused on business to business information distribution and transactions. In BCL's Engineering Services division, specialised design and installation of security firewalls and corporate LAN Internet connections is provided.

In February 1997, BCL was approached by one of its existing clients, a large New Zealand based brewery, which will be referred to as the Kiwi Brewing Company (KBC). KBC were investigating how to increase the profile of one of their premium brands of beer, referred to here as Kiwi Lager. Their internal marketing department had suggested to KBC management that a presence on the WWW would be an ideal way to increase the profile of Kiwi Lager both in New Zealand and overseas.

BCL became involved in discussions with Saatchi & Saatchi, KBC's advertising agency. Saatchi & Saatchi had conducted numerous advertising campaigns in newspapers, magazines and television for Kiwi Lager. KBC viewed Saatchi & Saatchi as being Kiwi Lager's 'brand steward', and therefore wanted them closely involved in the development process of any proposed system.

Saatchi & Saatchi had made the recommendation to develop a WWW site that would provide a soft branding theme for Kiwi Lager, in a similar way to other popular beer WWW sites such as that offered by Guinness in the United Kingdom. BCL's role was to provide the WWW technology to deliver and leverage the Kiwi Lager brand on the Internet.

The choice of BCL for this case study was based on the organisation's extensive prior experience with the development of WWW-based information systems. This presented an excellent opportunity to examine how a professional software development company was currently developing WWW-based information systems, with the view to comparing their practices with dissimilar organisations.

Two in-depth open-ended interviews were conducted with Ken Westfield, the Managing Director of BCL, during October 1997. The small size of the organisation, and the tight deadlines that the developers work to, resulted in access to the developers of the Kiwi Lager WWW site being restricted. However, Ken Westfield had worked closely with all BCL personnel involved on the project.

The interviews focused on the development style used in the Kiwi Lager WWW site, touching upon aspects of other projects in which BCL had been involved. Other anecdotal evidence was gathered through telephone conversations and email communication. The actual Kiwi Lager WWW site was also reviewed before and during the interviews, to provide additional clarity to the organisational events that occurred during the development of the project.

4.2.1 Kiwi Lager Functionality

The WWW-based information system developed by BCL was designed to capture the e-mail addresses of users who visited the site, and to reinforce the Kiwi Lager brand.

The structure of the Kiwi Lager WWW site is, in comparison to other sites developed by BCL, straightforward. This encourages the user to explore the site, and interact with information presented, making it easier to elicit the user's e-mail address.

The Kiwi Lager site is split into five areas: an anagram game, a history of the Kiwi Lager brand, a section on the All Blacks rugby union team, yachting, and a bulletin board section.

4.2.2 Development

The development of the Kiwi Lager WWW site for KBL occurred between March and June 1997. The site uses a unique blend of technologies. 'Cold Fusion' is used to provide dynamic HTML content, mirroring the functionality provided by Microsoft Active Server Pages. Microsoft Internet Information Server is used as the WWW server, providing content to either Microsoft or Netscape browsers. Microsoft SQLServer is used to store the content of the anagrams and record the e-mail addresses of users. These systems run under the Microsoft Windows NT 4.0 operating system.

After the initial approach in February by Kiwi Brewing Company to Beta Communications Limited, a number of workshops were held. These workshops involved input from personnel at BCL, KBC and Saatchi & Saatchi. The purpose of these workshops was to identify the functionality to be embodied in the system, and to gain additional ideas from people at KBC and Saatchi & Saatchi. A 'shopping list', as Ken Westfield described it, of ideas and concepts that people wished to see in the Kiwi Lager WWW site evolved from these workshops.

After deciding on the content of the WWW site, BCL moved to build the infrastructure and prototype the interface. Designing the architecture of the WWW site is seen as a key step by BCL, as they try to develop a working prototype for all their systems.

BCL identified the most appropriate architecture solution based on two criteria: the current architecture used by Kiwi Brewery Company, and the functional needs of the WWW site. Choices were made between the use of Microsoft ASP or Cold Fusion, Oracle or SQL Server, Windows NT or UNIX platform. Where possible the architecture of KBC was adopted, allowing for the possibility in the future of the site being managed by their IT department.

After finalising the architecture of the site in early March 1997, the actual system was designed and developed using a reusable prototype. This process was seen by BCL as being interactive with the client, and involved viewing the prototype on the live WWW server.

At no stage in the development of the system was any documentation created. Even during the workshops, only a brief list of the functionality of the system was produced. In the development of the Kiwi Lager WWW site there was no project plan, just broad project milestones that the developers aimed to meet. BCL's position on the absence of any documentation was explained by Ken Westfield in the following quote:

"The clients are not too concerned about documentation for the system, or project plans, just what the end product is, and not what it could but what it actually is."

Three prototypes were used during the development of the interface, over March and April 1997. The first prototype was considered to have been "pretty much there", with the following prototypes having minimal changes to fonts and colours. Ken Westfield attributed this to two factors: a good track record of "hitting the button" with the first prototype, and the client's lack of knowledge in the WWW environment.

The developers of the system have on average twenty years experience in the development of software, enhancing BCL's ability to provide consistent quality in delivering what the client wants, as Ken Westfield explained in the following quote:

"KBC was not really sure what they wanted. Often clients will ask for an Internet presence, but are not really sure what they want. So quite often we will develop something for them which they are initially very impressed with. From the initial 'wow' factor they usually come back with more tweaking and tuning."

The interface was composed using a basic text editor to create the HTML, and graphic design tools to modify artwork provided by Saatchi & Saatchi. Interestingly, the tools used in the design and development of the interface were limited. The tools offered by Microsoft and Netscape were considered by BCL as being too restrictive in creation of a WWW interface. A text editor provided much more freedom to the graphic designer, and allowed them to put their own 'flare' into each WWW site. The editor allowed the graphic designer to operate at the pixel level of the screen, and quickly make changes to the HTML code that underlies the WWW interface.

In May and early June 1997, the anagram game was created using the Java. By this time the other functional areas of the WWW site had been created, as they only involved the placement of text and imagery.

The complete WWW site for Kiwi Lager was completed in late June 1997. In total 300 hours was spent in developing the system, with approximately 50% of the time being spent in the design and coding of the anagram game. An exact breakdown of the time spent during the development is not possible, as BCL did not maintain any documentation.

4.2.3 Present Situation

The Kiwi Lager WWW site has been successfully operating for five months. BCL have identified additional functionality that may be incorporated into the WWW site for the future, but have yet to propose any changes to KBC. New anagrams are being introduced every two months, and BCL anticipate that the site will be re-worked over the next twelve months. A possibility is a re-designing of the main screen to maintain the freshness of the site.

In reflecting on the development strategy used in the project, Ken Westfield suggested future projects might have to take a different approach:

"We should use a different strategy in the future. But, given the time we had to get the system to market we couldn't have done it any other way. We had a project plan, but really we just had to get on with developing the system."

4.3 Cecil Project

Over the last two years a Computer Supported Learning system (CSL), also known as Cecil, has been developed by the Department of Management Science and Information Systems (MSIS) at The University of Auckland.

The Cecil project was chosen by the researcher for two main factors: ease of access to the project team by the researcher, and by the unique combination of technologies involved. The Cecil project provided an opportunity to explore how an organisation approached the development of a WWW-based information system embodying both client/server and WWW technology.

The research was conducted over the period of May through October 1997. The main avenue for gathering data came from formal open-ended interviews with the three main stakeholders in the project. A number of informal discussions were also held, aiding in the participants becoming more comfortable with the researcher.

Cecil forms part of an on going initiative by MSIS. The Cecil team is led by Associate Professor Don Sheridan and David White, and a development team of students, currently numbering three.

The foundation for the Cecil project has been built upon over the past six years. The system derived impetus upon the arrival of Don Sheridan to The University of Auckland, in 1995. Consequently, the research interests of Don Sheridan were combined with the ideas proposed by David White, thus forming the Cecil project.

During 1992 and 1993, David White created a Microsoft Access-based application called 'Grade Book', using a new data model. This application allowed MSIS lecturers and course co-ordinators to enter marks for students, print class lists, review student progress and print exam results.

Don Sheridan was interested in providing a system that could model a students behaviour while they took a test. In doing so, the department could achieve a means of understanding exactly what the student had done during the test. For example, inferences could be drawn regarding the delay patterns in answering questions or investigate the impact of question design or poor understanding of them.

In early 1995, Don Sheridan developed a functional specification for the Cecil system, with input from David White and Richard Vowles, a course co-ordinator. The interaction between David White and Don Sheridan led to the functional aspects of the Cecil system being discussed in David White's final year Information Systems Analysis and Design and Diploma of Business classes. In these classes, the students were encouraged to build prototypes of Cecil. These effectively became the first examples of the Cecil system.

4.3.1 Cecil Development Strategy

The Cecil architecture has developed over two phases over the period of March 1995 through to February 1997.

Initial discussions regarding the functionality and possible architecture decisions occurred in March 1995. A functional specification was prepared by Don Sheridan in parallel with the development of prototypes.

Prototypes were produced by final year students in David White's classes. During the evaluation of the student prototypes, both Don and David had a clear realisation that the WWW browser was going to be the delivery mechanism for the Cecil system.

Prior to the emergence of Cecil, The University of Auckland had been struggling to provide an interface for applications on multiple platforms that could be distributed throughout the university. Attempts had been made, but the issue had never really been satisfactorily resolved. The advent of WWW technologies had provided the opportunity to develop a WWW application that could operate across the entire university infrastructure.

The Cecil team were firmly committed to using WWW technology, but the main difficulty was in connecting to a database source. The prototypes developed by the students, especially the more advanced ones who used CGI, provided a source for comparison in deciding what was the best technology combination.

December 1995 saw comprehensive discussion between Kevin Bosch, Don Sheridan, David White and Richard Vowles. Much of the discussion involved getting the data model correct. The data model adopted aspects of, and expanded upon, the original created by David White. The data model was reworked constantly, often due to technological challenges posed by the WWW

The data model was seen as being a critical component of the system, as it formed a basis for the WWW interface and associated client/server applications. It is for this reason that much discussion was focused around ensuring the data model was acceptable, and could accommodate future change.

During December 1995, an important architectural decision was made involving the use of stored procedures in the database. This would later be changed to a Common Object Model (COM), however stored procedures were found to have a speed advantage over other technologies. The stored procedures held the business rules for Cecil, but the debate continued over whether to use them or a middle-tier approach.

Once a stable data model had been completed and the functionality decided upon, attention moved to the development of the interface. The interface was analysed and designed through a number of prototyping iterations involving three stages: the development of Microsoft PowerPoint slides, static HTML, and HTML with a connection to a database.

Following the design of the interface, the development process then focused on providing the additional functionality for the system. The prototype was reworked, with some elements being discarded, as it was very broad in its application, lacking depth in any of the functional areas.

The student interface was finally completed in March 1996, after which the developers concentrated on providing the client/server administration applications. At this stage in the development, both the student WWW interface and the client/server applications used a stored procedure architecture.

A turning point in the development of the system came in July 1996. David White, as in the previous year, was using students from his two classes to develop prototypes for Cecil. In his Diploma of Business class, one of the students was a full-time employee of Microsoft. His prototype used beta software from Microsoft, including Microsoft Internet Information Server (IIS) and Active Server Pages (ASP).

This prototype allowed the Cecil team to view the latest technology advancements in operation before they were released., leading to a re-evaluation of the system architecture, and a second generation of Cecil.

The Common Object Model (COM) was chosen to capture the business rules, forming an n-tier architecture, replacing the stored procedures. These architectural changes began to occur in December 1996.

The development team found that the move to a COM object model provided a great advantage over the previous architecture. The new architecture offered much greater reuse as both the WWW interface and the client/server applications written in Delphi, could take advantage of the COM objects.

In February 1997, the re-engineered Cecil system was delivered, despite the problems involved in the implementation of COM objects

4.3.2 The Future of Cecil

The Cecil system is currently being used in eight papers across a variety of departments within The University of Auckland.

As the 1998 academic year approaches, discussions are beginning regarding the direction that the Cecil project needs to take for the future. These discussions will focus around how the new technologies that have developed over the previous year can be adopted to improve the functionality and performance of Cecil. This may involve a migration towards the Microsoft Transaction Server, which will improve the ability to manage COM objects within Cecil.

Don Sheridan also commented on the need to maintain the 'freshness' of the WWW user interface. New advancements in HTML standards, and increased quality of other WWW pages that the students are familiar with, will eventually force the Cecil team to re-evaluate the interface:

"Yes we will probably have to look at the interface over the next six months. While it might be functional, it might not necessarily be 'fresh'. We see it in vehicles, they are continuously changing the look of the car each year. There should be no problem putting in some new logos."

The Cecil project is therefore in a transitional period, with the key drivers being new WWW technology and the need to improve the quality and functionality of the client/server applications.

5. Cross Case Analysis

This section provides a detailed analysis of the case studies discussed earlier, addressing the research questions that were presented in Section 3.1.1.

5.1 Research Question 1

The ASL developers did not feel they were following any methodology or structured approach.

"There was no methodology used. That's the bottom line frankly. To be honest, the bare-bones methodology we have is inadequate at best…we are still miles behind where other companies are."

These comments suggest that the underlying organisational standards and methodologies were not suited to the RAD approach taken in the project. Whilst standards were available to the developers in ASL's QMS system, they were not followed. As both Chris and Aaron commented, they just wanted to "get it going".

The technical challenges that the ASL developers faced appear to be the key driver in how the Internet-based Order Entry System was developed. The design and construction phases, that occurred concurrently, emerged as being ad-hoc. Problems with learning and using HTML, Java, and Perl resulted in technology driving the design of the system, and in some areas, the functionality. These technical problems forced the architecture of the system to be developed as the system was being constructed, with changes constantly occurring when a technical solution to a problem could not be found.

SOFTWARE PROCESS	METHODOLOGY	TECHNIQUES	TOOLS

SDLC	RAD	Prototyping	DBM tools
		E-R diagrams	Text Editors
		Object Orientation	Netscape Navigator
			Java SDK

Table 1 ASL WWW methodology summary.

The software process model used by BCL followed a similar process to that of the SDLC. The developers, however, were not aware they were following any process, but rather, they were conducting the development using “common sense”.

SOFTWARE PROCESS	METHODOLOGY	TECHNIQUES	TOOLS
SDLC	Ad-Hoc	Prototyping	Vi text editor
	(RAD)	Workshops	Cold Fusion
		E-R diagrams	Netscape Navigator
		Graphic Design /Composition	
		Marketing/ Corporate Alignment	
		Object Orientation	

Table 2 BCL WWW methodology summary.

A specific methodology was not used in developing the WWW-based information system for Kiwi Lager. However, the approach taken may be described as ad-hoc, and consisted of a strategy that BCL considered to be a core competency. The strategy focused around the use of a graphic designer and a developer using a prototyping philosophy. The short time frame of the Kiwi Lager development suggests that a form of RAD was used during design and development. The following comment by Ken Westfield summarises why no methodology was used:

“We have no need. Our projects are typically three or six months in duration, involving two people usually. We know what we need to do, so get the job done.”

The philosophy adopted during the design and construction phases was a delivery of ‘design’ and ‘content’. In providing these two aspects, BCL adopt a precarious strategy by using the latest technology in the creation of their WWW sites. Technology is therefore a key driver in the design and development of BCL’s WWW sites, and in particular, the Kiwi Lager site as Ken Westfield illustrates in the following quote:

“Technology definitely drives the analysis and design of our sites. You have to use the latest technology to be the best in the market.”

The software process model used in the development of Cecil followed a classical SDLC life cycle model. The process also adopted elements of the Spiral Model (Boehm, 1988), with an analysis of the risks involved occurring throughout the project.
The approach taken in the development of Cecil was viewed as being independent from the fact that the system would be developed using WWW technology. A “common sense” stepwise approach was taken due to the unknown technology problems that the WWW may have presented the developers.
As part of the “common sense” approach used, the Cecil development team attempted to adopt industry standards where possible. In the case of the WWW interface for the students, a structured approach was taken to its design in an effort to maintain consistency. The developers did not consult any WWW-specific guidelines for the development of the interface, but used a combination of feedback from the users and standards adopted from the windows environment. The drive to follow standards is illustrated in the following quote from Don Sheridan:

“If there was a convention that we could use, then we were more or less interested with sticking with that convention.”

SOFTWARE PROCESS	METHODOLOGY	TECHNIQUES	TOOLS
SDLC	Ad-Hoc	Prototyping	Vi text editor
		E-R diagrams	ERWin
		Object Typing	Netscape Navigator
		Object Orientation	Rational Rose

Table 3 Cecil WWW methodology summary.

5.2 Research Question 2

The development process and methods used in the Internet-based Order Entry System offered similarities to that of classical systems developed by ASL. The approach taken by ASL in the analysis of the data requirements was conducted independently of the requirements of the interface, and used modelling techniques that had proven successful in previous projects.

However, a number of key differences were apparent. These involved technology, interface and security. Technology was the driving force in the development of the Internet-based Order Entry System, not only in the use of WWW technologies but also in connecting the Sybase database to other legacy systems. The developers had no experience in using any WWW technologies. In previous client/server development, they had become accustomed to using one or two technologies in delivering the system, but this WWW system required the use of up to five different technologies

The ASL developers found the interface to be a key feature needing additional attention. The developers discovered the HTML language limited their ability to provide some of the functionality requested in the GBSL functional specification.

"It is a different medium. People expect different things from the WWW. In a windows environment you can get away with popping up a dialogue, without worrying about what a user might do to it."

The approach taken by BCL in developing the Kiwi Lager system is no different from that used in other client/server applications. Ken Westfield described the approach as being one of "common sense".

This "common sense" approach would involve prototyping to a "sensible stage", where the client was happy with the system. In the Kiwi Lager WWW site, three different prototypes were presented, as Ken Westfield explains in the following quote:

"We ask the client if they are happy with the system to ensure we are on the right track. We will develop up to a certain point, which is often the home page, navigation buttons and imagery. We ask if it has the correct functionality…that is 'common sense' development."

However, key differences are evident in the interface design and the time available for the project. At BCL the graphic designer is regarded as a key factor in the design of their WWW sites, giving the organisation a competitive edge within the New Zealand market.

BCL have identified that 'design' and 'content' are two key factors in an interface for any WWW-based information system. A graphic designer with information technology skills is able to provide both with their own artistic creativity. This is in contrast to client/server applications, where the Windows environment restricts the creative nature of the designers, forcing them to follow either explicit or implicit user-interface guidelines.

BCL have been using a form of RAD in the development of their systems. Whilst some of the developers had used this approach in their client/server systems, Ken Westfield had the view that development for the WWW was even more pressured. The BCL approach could be described as frantic, or Frantic Application Development (FAD) (Yourdon, 1996).

The changing technology in the WWW environment contributes to the FAD development process use by BCL. This is in contrast to client/server applications that the BCL developers believed had a more stable technological base.

The development of the WWW component of Cecil was approached in a similar manner to that of the client/server interfaces. The development of the COM object layer in 1996 enabled the developers to re-use this code in both the client/server applications and the WWW interfaces.

The main difference arose in the approach used in the design of the interface. A functional specification was created for both the client/server applications and the WWW interface, but the initial focus was placed on the WWW technologies, as David White described:

"The only difference is what you can deliver on WWW. It is less from a user interface richness point of view. You don't have the windows 'widgets' available to you. The possibilities are much less. I would ask for a certain functionality, but found it could not be achieved on the WWW."

This quote is interesting in that the WWW could be said to offer more in the way of user interface richness through the use of multimedia, not less. The difficulties in designing the interface appear to have arisen through a lack of knowledge, or possibly a lack of maturity, in the technologies being used.

5.3 Comparative Analysis

The varied organisational backgrounds presented an opportunity to review any deficiencies that may exist in the development processes used for WWW-based information systems.

By comparing the three case studies it can be seen that deficiencies exist in the areas of interface design, technology, guidelines and documentation.

The developers in all three case studies expressed the importance of the interface for WWW-based information systems, especially in the BCL case study. The WWW is a multimedia environment, and therefore the need for a graphic designer is much more apparent. There is a need to keep the interface fresh and appealing to the user, especially in the area of corporate branding, like the Kiwi Lager WWW site. Users expect a high quality of interface design; one that is attractive and invigorating to view.

However, the scarcity of skilled resource has led to the development of WWW-based information systems by developers who lack graphic design skills. In both the ASL and Cecil cases, the developers expressed the importance of graphic design in the interfaces of their WWW-based information systems

Without a graphic designer, the developer must rely on experience, often following the familiar guidelines for windows environments. This is inadequate, as users are becoming increasingly more articulate in their understanding of interface design due to the number of examples they are able to view on the WWW.

In all three cases, technology was presented as causing a problem in the design of the WWW-based information system. It appears that a deficiency exists in the management of new technology for the WWW.

While it is true that the same problem may exist with classical client/server applications, the problem is more prominent for the WWW environment. New technology is appearing on the market for the WWW at an ever increasing rate. It is conceivable that with each new project that the WWW developer is presented, similar technological and knowledge barriers will be experienced.

With technology for the WWW moving so rapidly, developers will tend to design for the present, using a "just get it going" scenario. Such a situation was prominent in the ASL case. Designing for the future became less of an issue, as the developers were focused on ensuring the system would work, rather than trying to accommodate the technological advances that may occur in the future.

In the Cecil project, the focus was on working towards standards. However, no specific guidelines exist for the development of WWW-based information systems, other than guidelines for the appearance of the interface. The BCL case has shown that successful implementations do not need to follow WWW interface guidelines. Additional guidelines may be required to enable developers who lack graphic design skills to embody creativity into their interface design.

The shortened time frames in which WWW-based information systems are needed, has led to a culture of FAD development, as outlined by Yourdon (1996). By examining the three case studies, it appears that the technical challenges the developers encounter during development leads directly to an absence of documentation.

5.4 Case Analysis Summary

While the WWW-specific methodologies presented earlier were neither known nor used by the organisations, the rapid movement of technology suggests that they may be inappropriate. This implies that even the new methodologies proposed for the WWW are not keeping pace with technology, as alluded to by Wynekoop and Russo (1997). The WWW-specific methodologies focused on the support for hypertext in WWW sites, but the case studies have shown that technology has moved beyond this basic WWW technology.

The analysis supports suggestions in the literature that technology is the current focus within the WWW community (December, 1996; Morrel, 1997), resulting in a lack of guidelines for addressing this problem. Furthermore, the analysis has revealed that changes have occurred in the development methodologies used by organisations operating in a technologically dynamic environment, with a move toward a FAD development strategy (Yourdon, 1996).

However, whether these changes in the development methodologies are seen as fundamental is open to interpretation (Wynekoop and Russo, 1997).

6. Conclusions

The analysis of the three cases has illuminated additional areas of concern that were not otherwise part of the initial focus of the study, resulting in a number of implications for researchers and practitioners.

This research indicated that traditional methodologies are being adapted within organisations, to accommodate the needs of the WWW environment. However, a limitation of the study was a narrow in-depth focus on three organisations. It may be fruitful to examine the development of WWW-based information systems from a wider perspective, using a survey of current practises as suggested by Wynekoop and Russo (1997). This would provide a contrast to this study, which took a historical perspective in the development of a WWW-based information system within an organisation.

This study also involved an examination of three organisations that had developed dissimilar systems. The WWW-specific methodologies presented earlier focused on supporting the development of systems that used hypertext extensively. It may be preferable to investigate the development of WWW-based information systems that use similar technologies, so that the impact of technology can be assessed.

The case analysis has highlighted a number of implications for practitioners. The practitioner should review the issues with respect to their development environment, as the implications arose from the organisational context of only three case studies.

Pressure arising from the constant introduction of new technology has introduced FAD approach into organisations developing WWW solutions. It appears that the FAD approach will become more prevalent as organisations become increasingly aware of the strategic value of a WWW presence.

In summary, the analysis of the development of WWW-based information systems within an organisational context has exposed many issues resulting from the complex interaction of technology and organisational culture.

The analysis of methodologies used within organisations is often subjective. In order for methodologies, tools and techniques to evolve in such technologically dynamic environments, research into the current software development practices must continue.

7. References

[