THE USE OF SYSTEMS DEVELOPMENT METHODOLOGIES IN WEB-BASED APPLICATION DEVELOPMENT IN SOUTH AFRICA

Martin Taylor, B.Com, Hons B.Com

Dissertation submitted in partial fulfillment of the requirements for the degree Master of Commerce at the Potchefstroom Campus of the North-West University

Supervisor: Dr. H.M. Huisman

November 2006
THE USE OF SYSTEMS DEVELOPMENT METHODOLOGIES IN WEB-BASED APPLICATION DEVELOPMENT IN SOUTH AFRICA

M. TAYLOR
Firstly I would like to thank Dr. Huisman for the guidance and support she has given me not only during this study, but in my Hons year as well. I’ll never forget our coffee adventures, and small talk in between work sessions. The extra support given this year with my studies changing to part-time is really appreciated.

To my parents, thank you for all the love and support during my years of study. Without all the opportunities you have presented me while I was studying, this would never have been possible. I love you dearly.

To all my friends and the one close to me, you know who you are, thank you for all your support during the last 2 years.

And to my old colleagues at TFMC, and new ones at SAP, thanks for the extra motivation and help these last few months.

Thank You.
This study investigated the use of systems development methodologies in Web-based application development in South Africa. Web-based systems differ from traditional information systems by integrating different media for knowledge representation and utilizing hypertext functionality. By doing this, Web-based systems not only support creation, integration, analysis, and distribution but also storage and transfer of knowledge of business transactions within a structured information system.

There are numerous methodologies available to develop Web-based systems. In this study five of these methodologies were discussed. The methodologies include Web IS Development Methodology (WISDM), Internet Commerce Development Methodology (ICDM), Web Engineering, Extreme Programming and the Relationship Management Methodology (RMM).

In this study a qualitative research approach was followed. Case studies were done on three different organizations in the South African marketplace. Semi-structured interviews were used for data collection at each organization. The interviews were transcribed, and the data were analyzed using content analysis and cross-case analysis. One of the main goals of this research was to determine “how” system development methodologies are used in practice to develop Web-based systems, and to what extent it is used.

The research pointed out that those organizations who participated in this study in South Africa mainly use in-house developed methodologies to develop Web-based systems, and that these organizations adhere strictly to their methodology. The main reasons organizations choose to use methodologies are that methodologies aid in the delivery of a better quality Web-based system, and also act as a good project management mechanism within the organization.

Keywords: System development methodologies, Web-based systems, Methodology, Web IS Development Methodology, Internet Commerce Development Methodology, Web Engineering, Extreme Programming, Relationship Management Methodology.
Die doel van hierdie studie is om die gebruik van stelselontwikkelingmetodologieë om Web gebaseerde stelsels te ontwikkel in Suid Afrika te onderzoek. Web gebaseerde stelsel verskil van tradisionele stelsel deurdat dit verskillende media gebruik om 'knowledge presentation' en 'hypertext' funksionaliteit gebruik. Deur dit te doen, ondersteun Web gebaseerde stelselontwikkeling, integrasie en distribusie asook oordrag van kennis binne 'n geslukte stelsel (Kaiser, 2000).

Daar is al klaar 'n paar stelselontwikkelingmetodologieë beskikbaar vir die ontwikkeling van Web gebaseerde stelsel. Dit sluit in Web IS Development Methodology (WISDM), Internet Commerce Development Methodology (ICDM), Web Engineering, Extreme Programming en die Relationship Management Methodology (RMM).

In hierdie studie word 'n kwalitatiewe navorsings benadering gevolg. Saak ondersoeke was toegepas op drie besighede in die Suid Afrikaanse mark. Half-gestruktueerde onderhoude vir data kolecisie was onderneem by die besighede. Die data was getranskribeer en gebruik vir kruis-saak analisering op die verskillende sake. Eén van die hoof doele van die navorsing is om ondersoek in te stel oor "Hoe" stelselontwikkelingmetodologieë gebruik word om Web gebaseerde stelsels te ontwikkel.

Befindings van die studie het gewys dat net een van die vyf stelselontwikkelingmetodologieë wat bespreek is amper voldoen aan die karakteristieke wat verband hou met Web gebaseerde stelsels nie. Die navorsing het getoon dat besighede huidige stelselontwikkelingmetodologieë aanpas om in hul besigheid konteks te pas. Die hoofrede dat besighede stelselontwikkelingmetodologieë gebruik is omdat die metodologieë help om beter kwaliteit stelsels te lever, en om te help met projek bestuur binne die besigheid.
CONTENTS

Chapter 1
Introduction
1.1 Problem Description ......................................................... 1
1.2 Research Goals ............................................................. 3
1.3 Research Method .......................................................... 3
1.4 Outline of the study .......................................................... 4

Chapter 2
Literature Survey
2.1 Information systems and the IT sector .................................. 5
2.2 Information systems development methodologies ................ 6
2.3 Web-based system development ....................................... 10
2.4 Summary ......................................................................... 14

Chapter 3
Web-based Systems Development Methodologies
3.1 Web IS Development Methodology (WISDM) ..................... 15
  3.1.1 Introduction ................................................................. 15
  3.1.2 Using the Multiview Framework .................................... 16
  3.1.3 The WISDM matrix ...................................................... 17
  3.1.4 The dynamics of WISDM .............................................. 23
3.2 Internet Commerce Development Methodology (ICDM) ........ 24
  3.2.1 Introduction ................................................................. 24
  3.2.2 Overview of ICDM ....................................................... 26
  3.3 Web Engineering: A new discipline for the development of Web-based systems ............................................. 32
  3.3.1 Introduction ................................................................. 32
  3.3.2 Framework for development of successful E-Business systems ................................................................. 34
3.4 Extreme Programming (XP) .............................................. 40
  3.4.1 Introduction ................................................................. 40
  3.4.2 The Basics ................................................................. 41
  3.4.3 Applying Practices ...................................................... 43
  3.4.4 Adopting XP ............................................................. 47
3.5 The Relationship management Methodology (RMM) ............ 48
  3.5.1 Introduction ................................................................. 48
  3.5.2 The RMM methodology .............................................. 51
  3.5.3 Conclusion ................................................................. 55
3.6 Summary ......................................................................... 56

Chapter 4
Methodology Evaluation
4.1 Methodology Comparison .................................................. 57
4.2 Methodology Comparison according to proposed framework .. 62
4.3 Theoretical evaluation of selected system development methodologies for their suitability to develop Web-based systems ......................................................... 66
4.4 Summary ......................................................................... 72
Chapter 5  
Research Method

5.1 Research approach ...................................................................................... 73
5.2 Research methods ................................................................................... 75
5.2.1 Standard Bank South-Africa .......................................................... 78
5.2.2 Working Webs ............................................................................ 79
5.2.3 Fourier Approach Pty (Ltd.) ........................................................... 80
5.3 Data collection ......................................................................................... 81
5.4 Data Analysis .......................................................................................... 87
5.5 Confirmation of findings .......................................................................... 90
5.6 Summary ................................................................................................ 91

Chapter 6  
Results

6.1 Findings for Standard bank ................................................................. 92
6.1.1 The Team .................................................................................. 92
6.1.2 The Project ............................................................................... 93
6.1.3 System Development Methods used ......................................... 93
6.1.4 Methodology influence .......................................................... 95
6.1.5 Summary .................................................................................. 96
6.2 Findings for Working Webs cc .............................................................. 97
6.2.1 The Team ................................................................................ 97
6.2.2 The Project ......................................................................... 97
6.2.3 System Development Methods used ................................... 98
6.2.4 Methodology influence ......................................................... 99
6.2.5 Summary .............................................................................. 99
6.3 Findings for Fourier Approach Pty (Ltd) ................................................ 100
6.3.1 The Team ............................................................................... 100
6.3.2 The Project ......................................................................... 100
6.3.3 System Development Methods used ................................ 101
6.3.4 Methodology influence ....................................................... 102
6.3.5 Summary ............................................................................ 103
6.4 Results of cross-case analysis ............................................................... 103
6.4.1 Context ............................................................................... 104
6.4.2 System development methodology use ................................ 105
6.4.3 Influence ............................................................................. 106
6.5 Summary ............................................................................................. 108

Chapter 7  
Findings and Contributions

7.1 Research contributions ............................................................................ 109
7.2 Limitations of the study ........................................................................ 112
7.3 Future work .......................................................................................... 113

References .................................................................................................. 114

Appendix A .................................................................................................. 120
Appendix B .................................................................................................. 125
Chapter 1

Introduction

1.1 Problem Description

A summarization of data collected from various organizations concluded that the average completion time for the development of an information system varies between 18 months and 5 years. Additionally, the CHAOS Report study (1995) revealed 68% of projects overrun their schedules, 65% exceed calculated budgets, 75% are suspect to major redesign after implementation and 35% of organizations have at least one runaway project (CHAOS Report, 1995). Information system development can be seen as a difficult task, and some believe that the solution is in the form of a more professional approach to development. One area that organizations have invested in is that of system development methodologies.

Earlier it was assumed that the Internet and Web was a means of communicating and sharing information across a widely dispersed audience (Fitzgerald et al., 2002). Since then this trend has changed from governments to multinational companies to one-person start-ups. The new focus being shifted on E-commerce, which is increasingly viewed as a key business modality of the future (Murray and Allison, 2003). E-commerce can be explained as a process of buying and selling products, services, and information over the Internet, Intranets, Extranets, and E-mail. In 2001 the Census Bureau in the US highlighted that national retail sales increased by 3.1% from the previous year, while at the same time, the online E-Retail sales grew by 26.8% (Carton, 2003). It is important to note that the organization needs to market the site, and the success of an E-commerce site depends heavily on the quality of the software applications that form its basis. This research will investigate whether or not the use of systems development methodologies has an influence on the quality of the produced Web-based system.

As the use of the Web moved from the static display of information to real-time interactive applications, the design and development of these Web-based systems has not only increased in complexity, but also present a lot of challenges due to the rapidly changing environment of the Web and E-commerce sector. The main problem confronting business and Web developers alike are compressed time schedules, having to handle more and larger build-or-buy decisions, and having to incorporate
and interface to legacy or hosted systems displaying various capabilities (Albertyn, 2002).

It is generally assumed that systems development methodologies can help to address the problems associated with systems development. However, Web-based system's development differs from that of traditional systems development. With traditional information systems, focus is mainly placed on accessing data directly from the source, reporting, and analyzing this structured data. Most of this is relevant to the business transactions and processes which the system covers. Web-based information systems go beyond this functionality by uniting different media for knowledge representation, and by including hypertext functionality. By doing this, Web-based systems support not only the creation, unification and analysis of structured information related to business transactions and processes, but also the storage and transfer of knowledge associated with this (Strauch and Winter, 2002).

Most of the systems development methodologies today were developed to address traditional systems development, and are believed to be not suitable for Web-based application development. The problem with classical heavyweight development methodologies is that they are geared more towards bottom-up building using a rigid planning approach. They are also too time consuming, and battle to cope with Web-based system requirements that are constantly evolving. Although some Web-based systems development methodologies exist that can be used to address these problems, e.g. WISDM, ICDM, e-RAD, Web-engineering, and webXstream, practitioners are not using them (Barry and Lang, 2001).

Studies that focus on how Web-based systems development methodologies are used to develop Web-based systems in South Africa are very limited. Two recent studies have been done by Huisman (2000) and Kalanjee (2006). The main focus of these studies was on the use and effectiveness of systems development methodologies in general and not on Web-based SDM specifically. Clearly there is a need for further research in this area.
1.2 Research Goals

The general goal of this study is to investigate how Web-based applications are developed in South Africa. Although formalized web application development methods have been proposed by the academic community, previous studies concluded that not all practitioners are using them (Barry and Lang, 2001).

This research will aim to address the following specific research goals:

1. Analyze and summarize a selection of different Web-based system development methodologies available for development.
2. Compare these system development methodologies according to a framework of Avison and Fitzgerald (2003).
3. Evaluate theoretically the selection of system development methodologies for their suitability to develop Web-based systems.
4. Describe the context in which Web-based systems are being developed.
5. Describe how system development methodologies are used in practice to develop Web-based systems, and to what extent it is used.
6. Describe the influence of system development methodologies on the quality of Web-based systems.

1.3 Research Method

This study will investigate companies that develop Web-based systems in their daily activities. These Web-based systems could range from an E-commerce sales or trading perspective, to a database server connecting individual companies through a Web environment. This study will follow the qualitative research approach. One of the main goals is to determine "how" system development methodologies are used in practice to develop Web-based systems, and to what extent it is used. This research will follow a qualitative research approach for the prevailing characteristics associated with it, as described in chapter 5.

The multiple-case study research method will be used for this study. A case study will be created for each of the three organizations visited. For data gathering, semi-structured interviews will be used. The same questions will be used at each organization. Questions will be divided into two parts, one part consisting of questions more aimed at the business and managerial side, and the other part aimed
at the developers and programmers. This will aid in getting a picture of the organization as a whole, and not just the applicable development team.

After the data has been collected, content analysis will be done for which the cross-case analysis method will be used. Cross-case analysis can aid in identifying shared goals, procedures and results among the selected body of research focusing on system development methodology usage if applicable. A tool that will be used in analysis is Atlas.Ti®, which is a qualitative data analysis application. Atlas.Ti® facilitates many activities used in text analysis and interpretation. It uses tools to handle, extract, compare and utilize meaningful segments of large data amounts in a user friendly way.

1.4 Outline of the study

Chapter 1: Introduction
This chapter gives a brief introduction in to the problem description, goals of the research, and the research method that will be used.

Chapter 2: Literature survey
In this chapter system development methodologies are discussed, as well as an introduction on Web-based system and their relevant characteristics.

Chapter 3: Web-based System Development Methodologies
In this chapter five Web-based system development methodologies will be discussed.

Chapter 4: Comparison Frameworks
The five methodologies discussed in the previous chapter will be compared by using a specified framework. A theoretical evaluation of these methodologies will be performed to determine their suitability for Web-based systems development.

Chapter 5: Research Method
The qualitative research method used in this study will be described.

Chapter 6: Research Findings
The findings derived from the case studies will be discussed in this chapter. Content analysis and cross-case analysis will be performed on the data collected in chapter 5.

Chapter 7: Conclusions and Discussions
A brief summary is given on this study, as well as a discussion on the limitations and future work associated with this study.
Chapter 2

Literature Survey

In this chapter software development will be discussed in general. There is a brief introduction into methodologies, including objectives, advantages and disadvantages. The differences between traditional and Web-based systems are discussed. A few common characteristics associated with Web-based systems are highlighted.

2.1 Information systems and the IT sector

Heading into the 21st century, information systems are becoming more complex than ever thought possible. Information systems are becoming highly technologically advanced, placing a bigger importance on how essential information systems are to an organization’s survival. Laudon & Laudon (2002) suggest four factors that have made information systems strategically important:

- **The Global economy**: Foreign trade in the United States represents more than 25% of economic activity. To be successful in the global market, organizations need powerful information systems to support this activity.
- **Industrial transformation**: Services are information and knowledge intensive, and many jobs rely on creating and distributing information.
- **Business transformation**: New business organizations have flattened organizational structures. These businesses assign resources around the customer's requirements. They also aim to deliver mass-customized products and services.
- **Digital firms**: E-businesses conduct relationships with customers, suppliers, and employees and are able to sense and respond to changes in their environment.

An information system is a set of interacting components. It consists of people, procedures and technologies. These components can be combined to collect, process, store and distribute information (Fitzgerald et al., 2002). This information can then be used to support the control and decision making by management in the organization.
2.2 Information systems development methodologies

In 1994 'The Standish Group' did research on information and software system development and the success and failure rates thereof. Their research concluded that in the United States alone, companies spend more than $250 billion each year on IT application development of approximately 175,000 projects. A staggering 31.1% of these projects will be canceled before they ever get completed. Further results indicated that 52.7% of projects will cost 189% of their original estimates. In 1995 American companies and government agencies spent $81 billion for canceled software projects. These same organizations will pay an additional $59 billion for software projects that will be completed, but will exceed their original time estimates (Carton, 2003).

The most important aspect of the research was discovering why projects failed. To do this, The Standish Group surveyed IT executive managers regarding their opinions on why projects succeed. The factors are summarized in table 2.1. The percentages on the right hand column highlight how important management feels the applicable factor is. The three major factors for a project to succeed are user involvement, executive management support, and a clear statement of requirements. There are other success criteria, but with these three elements in place, the chances of success are much greater. Without them, chance of failure increases dramatically.

<table>
<thead>
<tr>
<th>Project Success Factors</th>
<th>% of Responses</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. User Involvement</td>
<td>15.9%</td>
</tr>
<tr>
<td>2. Executive Management Support</td>
<td>13.9%</td>
</tr>
<tr>
<td>3. Clear Statement of Requirements</td>
<td>13.0%</td>
</tr>
<tr>
<td>4. Proper Planning</td>
<td>9.6%</td>
</tr>
<tr>
<td>5. Realistic Expectations</td>
<td>8.2%</td>
</tr>
<tr>
<td>6. Smaller Project Milestones</td>
<td>7.7%</td>
</tr>
<tr>
<td>7. Competent Staff</td>
<td>7.2%</td>
</tr>
<tr>
<td>8. Ownership</td>
<td>5.3%</td>
</tr>
<tr>
<td>9. Clear Vision &amp; Objectives</td>
<td>2.9%</td>
</tr>
<tr>
<td>10. Hard-Working, Focused Staff</td>
<td>2.4%</td>
</tr>
<tr>
<td>Other</td>
<td>13.9%</td>
</tr>
</tbody>
</table>

Table 2.1 Success profiles and responses from IT management (Standish Group, 1994)

Fitzgerald (1998) has summarized the evidence for a software crisis and has concluded that the average completion time for the development of an information
system varies between 18 months and 5 years; 68% of projects overrun their schedules; 65% exceed calculated budgets; 75% are suspect to major redesign after implementation and 35% of organizations have at least one runaway project.

Information system development can be seen as a difficult task, and some believe that the solution is in the form of a more professional approach to development. One area that organizations have invested in is information systems development methodologies. A methodology has been defined by Avison & Fitzgerald (2002:20) as:

“A systems development methodology is a recommended means to achieve the development, or part of the development, of information systems based on a set of rationales and an underlying philosophy that supports, justifies and makes coherent such a recommendation for a particular context. The recommended means usually includes the identification of phases, procedures, tasks, rules, techniques, guidelines, documentation and tools. They might also include recommendations concerning the management and organization of the approach and the identification and training of the participants.”

Huismann and livari (2006:32) define a systems development methodology as a combination of the following:

- A systems development approach. This approach includes the philosophical view on which the methodology is built. It includes a set of goals, guiding principles and beliefs, fundamental concepts, and principles of the system development process (livari et al, 1998).
- A systems development process model. A process model can be seen as a representation of the sequence of stages through which a system evolves.
- A systems development method. A method can be seen as a systematic way of managing at least one complete phase of systems development. This includes a set of guidelines, activities, techniques and tools which are based on a particular philosophy and the target system (Wynekoop, 1993).
- A systems development technique. Development techniques can be defined as procedures used to perform a development activity.

1 In the academical field there has been significant debate regarding the appropriate use of the term method or methodology. In this study the author will use the term methodology throughout.
Furthermore, a methodology is also more complex than a collection of these techniques and tools, and is usually based on some 'philosophical' view (Avison and Fitzgerald, 2006). Methodologies mostly differ in the techniques and tools used in the phases of development, but sometimes the differences can be seen as more fundamental. Some methodologies follow the soft systems approach which emphasizes the human aspects in developing information systems. Other methodologies aim at a more scientific approach, other at a pragmatic approach, and some try to automate the whole project development cycle (Avison and Fitzgerald 2003).

Techniques and tools feature in every methodology. A technique is a way of doing a certain activity in the applicable phases within the methodology. Some methodologies may use many of these activities in their approach. These techniques make use of certain tools to aid in the process. These tools normally represent some of the artifacts used in information systems development.

Avison and Fitzgerald (2003) identified a few of the different objectives of methodologies as:

- To record accurately the requirements for an information system.
- To provide a systematic method of development so that progress can be effectively monitored.
- To provide an information system within an appropriate time limit and at an acceptable cost.
- To produce a system that is well documented and easy to maintain.
- To provide an indication of any changes that need to be made as early as possible in the development process.
- To provide a system that is liked by those people affected by that system.

Vidgen et al. (2002) justify the advantages of adopting a methodology as follow:

- The subdivision of a complex process into manageable tasks.
- Facilitation of project management and control. One role of management is to manage risk and uncertainty.
- Purposeful framework for applying techniques.
- Economics – skill specialization and division of labor.
- Epistemological – a framework for the acquisition and systemization of knowledge. A methodology should promote organizational learning.
• Standardization – interchangeability of developers, increased productivity and quality.

However, Avison and Fitzgerald (2003) also identified problems with methodologies. There is a fundamental issue about what is a methodology, and this led to certain definitional anomalies. These anomalies created a tendency for methodology authors to engage in ‘method wars’. Some methodologies have few or weak conceptual and empirical foundations. Those with conceptual support tend to be rooted firmly in a scientific and engineering paradigm. Engineering was possibly an appropriate reference discipline when development was concerned with hardware and software support for repetitive processes.

Avison and Fitzgerald (2003) also concluded that this can however be seen as a crude approach when dealing with social systems. It is also possible that the methodology becomes an end in its own right. A methodology helps to organize and frame problems more clearly, but it can also be a way of not seeing that the chosen methodology might not be relevant or needed. It is important to be aware of the dangers of focusing too much attention on selecting and using methodologies, where it actually might not be needed or be unsuitable.

Avison and Fitzgerald (2003) have highlighted some of the criticism associated with using methodologies to develop systems:

• It is believed that methodologies fail to deliver the suggested productivity benefits, because they do not reduce the time used to develop a project.
• Methodologies have been criticized for being over complex.
• Methodologies gather more requirements that might be legitimately required.
• Methodologies often require important skills in their use and applicable processes which might require users and end-users to learn additional skills.
• Methodologies use certain tools which might be difficult to use and do not render sufficient benefits.
• The methodology is not contingent upon the type of project or its size.
• Methodologies adopt one approach to the development of projects, which might limit the focus on underlying issues or problems.
• The methodology could limit changes to requirements during development.
• Methodologies could make a number of simplifying assumptions that could be incorrect.
The existence of a methodology standard could lead to an organization focusing all the attention on their methodology, and exclude the real needs of the project.

It is believed that by using a methodology, the development team wrongly does not have to give a lot of attention to the problem situation, and that the methodology will address the issue for them.

Methodologies focus insufficiently on social and contextual issues.

Some organizations have found it hard to adopt a methodology in practice.

It is perceived that by using a methodology it not necessarily results in a better system at the end of the day.

It is important to note that these advantages and disadvantages have an influence on the decision of the system development team to make use of a methodology for the development of their Web-based systems or not.

2.3 Web-based system development

Earlier it was assumed that the Internet and Web was a means of communicating and sharing information across a widely dispersed audience (Vidgen et al., 2002). As the use of the Web moved from the static display of information to real-time interactive applications, the design and development of these Web-based systems has not only increased in complexity, but also present a lot of challenges due to the rapidly changing environment of the Web and E-commerce sector.

Surveyer (2001) stated that the main problem confronting business and Web developers alike are compressed time schedules, having to handle more and larger build-or-buy decisions, and having to incorporate and interface to legacy or hosted systems displaying various capabilities.

In the following section the differences between traditional information systems and Web-based information systems will be discussed. Kaiser (2000) classifies Web-based systems from a business perspective as:

- Business Platform: For E-commerce business models like electronic auctions or process portals, certain Web-based systems become the backbone of their operations.
• Sales and purchase channel: For traditional business models certain Web-based systems are used to support an additional sales/purchase channel (E-commerce).

• Self service: In any business, certain Web-based systems can be deployed internally to decentralize selected support processes.

• Information management: Certain Web-based systems support the creation, integration, analysis, and distribution of information, particularly and the supporting of management processes.

While the first three classifications focus on the business transaction aspect side, the last classification characterizes Web-based information systems. Traditional information systems focus on querying, reporting, and analyzing structured data related to business transactions. Web-based information systems go beyond this functionality by integrating different media for knowledge representation and by hypertext functionality. By doing this, they not only support creation, integration, analysis, and distribution but also storage and transfer of knowledge of business transactions within the structured information system (Kaiser, 2000).

Web-based information systems also allow for more complex forms of information management than traditional systems. As a result, organizational, functional, and data views of business processes have to be specified during the conceptual design of the Web-based information system (Kaiser, 2000).

Liaw and Huang (2002) name a few popular characteristics associated with Web-based system's environments:

a) Web-based systems offer a multimedia environment. The information in Web-based systems can be simultaneously represented in any combination of media format, including text, image, graphic, sound, voice, and animation.

b) Web-based systems integrate various kinds of information and construct information bases. The multiple mixed-media nodes in a hypermedia system, such as the Web, can be instantly called up in a consistent manner.

c) Web-based systems support interactive communication. In general, Web users have full control over their own learning situations. This high level of interaction gives users dynamic control of information.
d) Web-based systems support networks to access information. A Web-based system goes beyond static Web pages and page linking. It creates truly interactive networks with information exchange between users and servers.

e) Web-based systems provide a cross-platform environment. A cross-platform highlights that Web systems can be executed independently on various computer operating systems. In the Web, information and resources from around the world can be accessed by anyone from anywhere in the world as long as he or she has a computer with an internet connection.

Along with these characteristics, Baskerville & PriesHeje (2001) also identified the following:

1) Web-based systems adhere to time pressure. Competitive pressures may mean that any advantage is short-lived and will be copied quickly. Therefore it is important to take advantage of any short-term gain to lead to more long-term advantages.

2) Web-based systems often have imprecise requirements. It is not infrequent that it is only on implementation and use that the real requirements are revealed.

3) Web-based systems rely on prototyping. The software prototype is the specification of requirements, not the thick paper report or specifications.

4) Web-based systems are released early and often. These ideas with others suggest early release and frequent re-release. Rapid application development is even more relevant to Web projects.

5) Web-based systems development supports parallel development. For example, database development can take place at the same time as the graphical design, and requirements analysis and design become hard to separate.

6) Web-based systems require a fixed architecture. Complexity needs to be tamed. A three-tier architecture where the business data, business logic, and the user interface are separated out, allows for team members to work in parallel with a degree of independence.

7) Web-based systems support the coding of a way out. If the Web-based system needs changes required urgently, to reduce downtime, it should be possible for the programmers to make fixes quick and easily.

8) Web-based systems focus on quality in terms of the view from its users. The question often arises as to whether software is developed to achieve high
quality, a quick product, or a cheap one. In some senses, quality has always been negotiable. In Web-based projects the overriding view of quality tends to be the customer perspective and experience, rather than by a defined and repeatable development process or a software product that survives an internal audit.

9) Web-based systems depend on good people. Web projects are completed under time pressure and typically in small teams where all members need to pull their weight. Key staff can make or break a project.

10) Web-based systems require structure. The old structures of system development, for example, keeping business analysts separate from software engineers, may be inappropriate to building applications in Internet time.

Surveyer (2001) states that the problem with classical heavyweight development methodologies is that they are geared more towards ground-up building using a rigid planning approach. They are also too time efficient and battle to cope with Web-based system requirements that are constantly evolving. More lightweight methods are coming into fashion because they are orientated to an “integrate and test” process. These methodologies are more relevant to cope with packaged solutions and components, because the third party themselves become part of the discovery, integration, and testing cycles (Surveyer, 2001).

Barry and Lang (2001) concluded from their research on Web and multimedia systems development that 24.6% of companies don’t use any methodologies at all. Although some Web systems development methodologies exist that can be used to address these problems, with examples that include WISDM, ICDM, e-RAD, Web-engineering, and webXstream, practitioners are not using them. Of those who use some type of methodology, three-quarters use an in-house variant as their principal methodology (Barry and Lang, 2001). An objective of this research will be to study methodology use for Web-based systems development within certain target companies in South-Africa. The study will aim at concluding whether any methodology or methodologies have been used, and if applicable, to what extent it has been used in South-Africa.
In this chapter information system development in general was discussed and focus was placed on why these information systems are important for organizations heading into the 21st century. The next aspect discussed was the usage of methodologies for the development of these information systems. A definition of methodologies was provided, as well as the advantages and disadvantages associated with methodologies.

After this, Web-based system development was discussed, as well as the difference between traditional information systems and Web-based information systems. Some of the characteristics associated with Web-based systems were also supplied to aid in understanding the differences of this class of system better.
Chapter 3

Web-based Systems Development Methodologies

This chapter will focus on some of the methodologies currently available for the development of Web-based systems. The following methodologies were chosen based on the availability of information:

- Web IS Development Methodology (WISDM), Vidgen et al. (2002)
- Internet Commerce Development Methodology (ICDM), Standing (1999)
- Web Engineering, Murugesan (1998)
- Relationship Management Methodology (RMM), Isakowitz et al. (1995)

For a more comprehensive list of methodologies and techniques used for the development of Web-based systems please refer to Lang (2006:282-284).

3.1 Web IS Development Methodology (WISDM)

3.1.1 Introduction

The first Web-based system development method that this chapter introduces is the Multiview framework for information system development. The methodology was developed by Vidgen et al. (2002) and information used for the rest of the WISDM discussion was sourced from this developer. WISDM utilizes this framework as a building stone in the Web IS development methodology. Multiview was designed to give an alternative from the trend that information system development followed focusing on engineering methods and disciplines. The fundamental assumption of Multiview according to Avison & Wood-Harper (1998) is that an information system methodology that relies mostly on an engineering approach and technical rationality is, by itself, an insufficient foundation for information system development.

The foundations of Multiview rest on the recognition that the needs of computer artifacts, organizations, and individuals must be considered jointly. Multiview is a framework for making the tensions in information systems explicit, to avoid distinctions between technical expediency and social justice. This aspect of negotiating between the organizational, technological and human aspects of
information systems development has established a central theme in the Multiview framework.

3.1.2 Using the Multiview Framework

Multiview can be used as a methodical guide for information systems mediation. It consists of a reflexive learning process. This process combines both the analyst and the situation with the methodology. Multiview is organized in three tiers (Vidgen et al., 2002:31):

a) General framework
b) Local, emergent methodology
c) Methods/Techniques

The Multiview framework provides a basis for the construction of a situation-specific methodology (figure 3.1). This methodology is the result of a commitment between the information system developers and the problem situation. This engagement then supplies information on the choice of methods and techniques to use on the applicable problem situation. Multiview can be seen as a simile that is interpreted and developed in a particular situation.

![Figure 3.1 The Multiview Framework (Vidgen et al. 2002:31)](image)
Within the framework there are methods that the developer can rely on to give insight into the problem. This insight can be obtained by using these methods to interact with the problem situation. The information system development methods are shown in the matrix in figure 3.2. The matrix presents methods and techniques. These are general tools that can be drawn upon in a specific situation. This leads to the particular people creating a local methodology in practice. This methodology is known as WISDM.

3.1.3 The WISDM matrix

The information system development matrix (figure 3.2) categorizes methods into two dimensions: socio and technical dimensions. These two dimensions can also be classified in either analysis or design. Hard or engineering based approaches in systems development focus on the right hand side of the matrix. This side concentrates on the requirements specification and its classification into a software model. The left hand side is more focused on the soft systems, people orientated approach. With this approach, both value creation and user satisfaction is intensified. The Multiview approach seeks a socio-technical solution. This solution gives balance to the left and right hand sides of the matrix and gives a more impartial approach.

![Figure 3.2 Information system development matrix (Vidgen et al. 2002:32)](image)

The WISDM matrix consists of five aspects:
- Organizational analysis
- Work design
- Information analysis
- Technical design
There is no sequence of ordering between the aspects within the methods matrix. Vidgen et al (2002:34) gives a clear example in explaining the different aspects by using the metaphor of a film camera. The focus of attention changes as the camera zooms in and out of an aspect. The camera can move from one aspect to the next. It can even revisit an aspect and found that the situation has changed. One particular aspect might be in focus and receive all the attention at a particular time, but the other aspects are in peripheral vision. They can still make their presence felt even though they might be out of shot. It is also possible to see all five aspects at the same time, but only by sacrificing the level of resolution. By contrast, it is possible to examine one particular aspect in great detail, but also at the expense of losing some of the context.

Each of the five aspects will be discussed in turn:

3.1.3.1 Organizational analysis (Value creation)

The traditional view of organizations can be seen as independent entities that seek to control their environment while engaging in conflict and competition with one another. An organization develops relationships through building and maintaining a strategy. Within this strategy a broad range of stakeholders include customers, employees, government, suppliers, labor organizations etc. The systems approach within organizational analysis is also seen as a subset of the broader aim of relationship maintaining.

a) e-Business Strategy

The underlying question for the information system development team should be how the project fits within the organization's business strategy. The project is part of a portfolio of information system development that in some way, align with the organization's business strategy. There are many different definitions of strategy, but Porter's (1980) definition is concise and to the point:

"Strategy is a broad based formula for how business is going to compete, what its goals should be, and what policies will be needed to carry out those goals. The essence of formulating competitive strategy is relating a company to its environment."
With Web-based development projects there is often a direct and tangible relationship between both information systems and strategy. Information system development may for instance have implications on the way the organization deals with customers and other partners, as well as having a direct impact on revenue flows. In particular, in some organizations the information systems are the very embodiment of the organization's business strategy.

b) System's thinking and SSM (Soft Systems Methodology)

System's thinking is particularly powerful in situations that are perceived as 'messes'. Here exploration and discussion are needed to define what the problem is, before rushing in and trying to solve it. A system is an entity that maintains its existence and functions as a whole through the interaction of its parts. The change in one part leads to change in all the other parts as well (Vidgen et al., 2002:83). These systems are also organized in a repetitive manner, and hierarchically. With this organization of systems within systems, all share the same systems mapping. It is important to note that with systems thinking, attention should be moved from the problem-solution approach, to a more relaxed recursive approach.

SSM was developed as a response to hard systems thinking and its failure to address 'messy' situations in which no clear problem definition exists. In soft systems thinking, thinking about the world is systemically, while recognizing that the perceived world is troublesome. SSM supports an enquiring process and it most commonly presented as a seven stage model (Checkland & Scholes, 1990). The stages are:

1. Situation considered problematic
2. Problem situation expressed
3. Root definition of relevant system
4. Conceptual models of system described in root definitions
5. Comparison of models and real world
6. Changes systemically desirable, culturally feasible
7. Action to improve the problem situation

For information, system developers could help hidden and unarticulated assumptions about the problem situation to be expressed. It may also help developers to think in terms of radical change and innovative new ideas. A
disadvantage might be that SSM has been criticized for weaknesses in addressing critical aspects of organizational change related to power and right of possession.

3.1.3.2 Work design
With traditional systems development methods in mind, developers tend to place a lot of emphasis on functional requirements. Another approach could be to specify these requirements in a diagram notation. Diagram examples include UML or Interaction diagrams.

Socio-technical approaches try to increase user participation. With this path a suitable fit between humans and technology is sought without trying to excessively adopt one of the two parties. With the coming of the Internet it can be probable that many of the users of the information system will be outside the organization. These users can occupy different roles such as customers, suppliers, partners, collaborators, investors etc.

There are two socio-technical approaches to information system design: ETHICS and participative design (Vidgen et al, 2002:101).

i) Socio-technical design with ETHICS
ETHICS (Effective Technical and Human Implementation of Computer-based Systems) is a socio-technical approach to information system design and implementation developed by Enid Mumford (1995). ETHICS aims to promote genuine participation. This genuine participation can be achieved by providing input to requirement specifications by users. There is also looked at an 'after-the-event' prototype evaluation (Vidgen et al, 2002:101). By following ETHICS, the user and technical designer identifies issues and objectives that the system has to deal with. Current practices that are followed, as well as the business structure may also need to be redesigned. The ETHICS approach places importance on the need to integrate people and tasks with technology and the business.

ii) Participative design
The participative design approach to computer applications can be summarized as follow (Vidgen et al, 2002:110):

- Computer applications should enhance workplace skills rather than degrade them;
• Computer applications are often used to improve productivity. These applications could also be used to improve the quality of the product.
• The computer system should support flexible work practices. The system must be designed to be under the control of the people who use them.
• The relationship between computers and work need to be addressed directly in design.

Participative design uses a structure that consists of four phases. In the first phase developers learn about the workings of an organization through interviews and user demonstrations. In the second phase all the attendees have the opportunity to contribute. Brainstorming sessions can be used to identify current problems or issues and categorize these accordingly. With the third phase current roles and tasks within the organization are described and new possibilities explored. The fourth and final phase embodies these ideas together and implements them in a mock-up prototype design.

3.1.3.3 Information Analysis
Analysis is often presented as ‘what’ the application will do, and design as ‘how’ it will be achieved. Analysis seeks to understand and represent the problem situation or domain. Two popular analysis and design approaches are defined by Vidgen et al (2002):

a) Structured systems analysis and design

Structured system analysis and design came out of the structured programming methods of the 1970s. It gained huge popularity in the 1990’s in the United Kingdom with SSADM (Structured System Analysis and Design Method). A common theme in structured methods is to picture the system from three perspectives:

• Data
• Process
• Dynamics

The principal models used in systems analysis are the entity relationship diagram, the data flow diagram and the entity life history.

b) Object-oriented analysis and design

Many benefits have been claimed for the object-orientated approach to systems development. These include reusability, reliability, scalability, faster
development, easier maintenance, and a more effective way of handling complexity. By using the same paradigm throughout the development cycle, the same notations can be used at different levels, making the object-orientated approach an attractive option.

There are many methods for expressing requirements. WISDM highlights one method that can be used for the analysis of information system requirements. This method is known as UML (Unified Modeling Language).

The core UML techniques used in information analysis are:

- Use cases
- Class diagrams
- Interaction diagrams
  - Sequence diagrams
  - Collaboration diagrams
- State transition diagrams
- Activity diagrams

3.1.3.4 Technical design
In some projects the development team might start by plotting a few ideas on a piece of paper. In other cases, the project team may be required to use specified technologies and approaches. This could be as a result of the standards that an organization needs to maintain or even from a favored technical platform of a senior manager. A problem with this restrictive approach can be that the project team may be forced to use inappropriate technologies. On the other side, if a project is given too much free reign, choices can be problematic. The organization might not have adequate infrastructural support to handle these new technologies. There are also plenty of technology platform decisions to be made before physical design can be complacent:

- What Database platform would the system require?
- Will a specific operating system be utilized?
- What would be required from a server side, and on which networks will data transfers take place?
- What license implications must be adhered to?
When beginning with logical design and modeling, low importance is set on the technical platform required. As design approaches a more physical side, the decision taken on technology platforms would play a greater role.

3.1.3.5 HCI
The design of the HCI (Human Computer Interface) has been a fundamental activity in the system development process. Web-based system's interface design differs from that of traditional systems. It requires a different mindset from the developer focusing more on graphical design than on just being creative with an interface layout. Graphic design is intensely visual. To build a web site that is visually appealing and practical for both customers and employees, it is clear that a fair amount of graphical design skills will be needed. According to Vidgen et al (2002) three options should be considered:

- Include an experienced web graphic designer in the project development team;
- Train a traditional IS developer in web graphic design;
- Buy a web template of the shelf.

In the end the most likely route is the appointment of a professional web designer to the IS development team.

3.1.4 The dynamics of WISDM
The multiple perspective approach described by Mitroff & Linstone (1993) can be used to inform the particular occurrence of Multiview under any set of circumstances. These perspectives form a part of the IS development matrix displayed in the Multiview framework in fig 3.1. Mitroff & Linstone argue that complex problem solving requires the application of as many disciplines, professions, and subdivisions of knowledge as possible, with of these employing their own different paradigm of thought. The idea of 'Multiple Perspectives' is used to describe these various ways of thinking. Three perspectives identified are:

a) Technical perspectives
b) Organizational perspectives
c) Personal perspectives

These perspectives can be used to emphasize how we look at a problem situation. By doing this the problem can be approached at from different viewpoints. The
advantage of this approach is that there is less importance placed on what the problem looks like at first glance. This provides a richer base from which investigations can be made into the complex problem situation. Any problem can be viewed from any perspective. Different perspectives may reinforce each other, cancel each other out, or operate in the dialectic mode. Choosing a particular perspective involves the investigator's ethical values and moral judgments.

Nearly all complex problem situations will inevitably require viewpoints from all three perspectives. It is therefore an essential aspect of IS development that the analyst develops self-awareness and becomes capable of thinking and acting on the joint basis of the three perspectives.

3.2 Internet Commerce Development Methodology (ICDM)

3.2.1 Introduction

The Internet Commerce Development Methodology (ICDM) is aimed at business-to-consumer application development and provides a guiding framework for developers. ICDM attempts to address the issues related to emphasizing a business focus, external focus, and speed of change that consumer applications development
requires. Information used throughout the rest of the ICDM discussion was mainly sourced from the developer of the methodology, Standing (1999).

ICDM views electronic business developments as organizational initiatives. With this following, ICDM takes into account the needs to address (Standing, 2001:549):

- Strategic issues
- Business issues
- Managerial disputes
- Organizational culture related problems
- Technical details of design and implementation.

ICDM relies on competitive analysis (SWOT analysis) to help shape the electronic business direction. The methods used for the development of a business strategy and defining requirements are intensely social in nature. Methods include Brainstorming and groups requirements sessions. Internet commerce should be a continually evolving feature of the organization, and strategy must be recognized as a socially constituted process, not static. Any methodology supporting this constituted process should aim to be interwoven with a dynamic learning environment.

ICDM is both a business analysis and systems development methodology. It takes into account the wider trends in the business world and society in its strategy development phase with the SWOT analysis. The changing profile of the consumer is important and user involvement is factored in at various points in the methodology.

It is important to note that ICDM provides a framework for developing Internet commerce. It is not a prescriptive methodology with a large amount of steps to be completed. It is a loose fitting framework for developing strategies and for the evolutionary development of Web-based systems (Standing, 2001:549). The phases of the ICDM methodology can be seen in figure 3.4.
3.2.2 Overview of ICDM

ICDM can be seen as a framework for the development of Internet Commerce in an organizational context (figure 3.5). ICDM provides both a management and development strategy. Both of these strategies are driven by needs of the business. ICDM gives particular attention to providing a business focus.
ICDM has the following components and features which are described next (Standing, 2002:156):

- Web management structure
- Strategy and business analysis development phase
  - SWOT analysis
  - Level of change
  - User involvement
- Meta-development strategy
- Analysis phase
  - Requirements techniques
  - Functional requirements framework
- Physical architecture framework
- Design phase
- Component implementation and evolution

Standing (2001:552-554) describes the following components and features in detail:

3.2.2.1 Web management strategy
ICDM suggests the development and management of e-business systems on three levels (figure 3.6). Web strategy can be seen as the development of the functional components of a web application. It is also perceived as an on-going task. The first level is a perspective on meta development and management. This provides a
framework for development. The second level is concerned with the development of the various components of the website. At both these levels the tasks need to be seen as being evolutionary in nature. This technique might help to cope with the inevitable changes that will have to be made later on in development. The third level in the management and development structure is concerned with developing and implementing the system. This includes technical development teams, analysts, content specialists and Web development consultants.

Figure 3.6: Web management and development structure (Standing, 2001:52)

3.2.2.2 Strategy and business development phase of ICDM

ICDM provides a strategic planning approach that considers which option is most appropriate for a given situation (figure 3.7). It draws upon Business Process Re-engineering (BPR) and Value Chain Analysis as its core strategic planning tools.

When deciding on a strategy for a business, business units or functional area, managers need to assess the organization's competitive situation. This involves assessing the organization and its environment, and is known as competitive analysis. A method which ICDM uses for competitive analysis is known as SWOT analysis. The competitive situation of the company is assessed by examining its Strengths (S), Weaknesses (W), environmental Opportunities (O), and Threats (T). This analysis will yield different results for every organization examined.

The strengths of the business can be found in the SWOT analysis examination. Internal strengths are features of the organization such as a streamlined administrative system, or technologically placid staff. The internal weaknesses can
be detailed in the same way. When looking at the wider environment, possible views to consider are economic, social, and technological trends that can be exploited. An example might be that new government legislation may create an opportunity for some organizations to reap a large amount of benefits.

Suppliers of products and services can market and sell directly to consumers with great ease. Companies that act as intermediaries in the distribution chain risk being by-passed by these suppliers. This could possibly lead to disastrous consequences for the intermediaries and is termed disintermediation. The businesses that are at most risk of disintermediation are those that do not significantly add value to the products and services that they are dispersing.

The scale and scope of the change should fall into one of three categories (Standing, 2002:157):

- Process change
- Process re-engineering
- Transformation

Process change is related to the enhancement or modification of an organizational process with for example the aid of the Web. Process re-engineering is the complete redesign of a process with the aid of the Web. Transformation is the radical change of a business leveraging Web technology.

![Organizational Web Management Team](image)

![Web site or Component Production Team](image)

![Analysis](image) ![Discipline Specialists](image) ![Programmers](image) ![Web Consultants](image)

Figure 3.7: ICDM strategic Planning Phase (Standing, 2002:155)
3.2.2.3 Meta-development strategy

When a company manages the development of a website it can employ a number of strategies. The strategy options available depend upon the amount of regulation or control that is desired, for both content and design. The strategy should:

a) Plan the entire site and regulate its distributed development in consultation with business units.

b) Plan the central part of the web site and allow business units the autonomy to develop their own neighborhoods.

ICDM suggests that the Web management team should decide on which option they would like to adopt (Standing, 2001:553).

3.2.2.4 User Involvement

Users of the system should be involved at various stages of the electronic business operations and be included in periodic reviews. Market research teams could be used to obtain information on what customers require and possible problems arising from using the Web. Customer input is essential at the strategy development and business analysis stages. Information on requirements can be obtained in group requirements sessions, telephone interviews or questionnaires. Customers can be involved in assessing design issues through the use of prototype Web systems. Customers should also be included in testing and evaluation of the Web site and can help in supplying the needed feedback. As strategy is likely to evolve through time, focus groups can be used to provide input through reviewing the current system and making recommendations (Standing, 2001:553).

3.2.2.5 Site and component development

Functional or divisional components of the internet system can be approached as discrete projects. When combining components in web applications it should be noted that implications could arise. Functional components can be for example a user having the option to question a database of products, or obtain details about customers for marketing purposes. ICDM suggests that a multi-disciplinary team is still required because any component of a Web site is still concerned with implementing business strategy, and not just technology.

3.2.2.6 Requirement analysis techniques

There are a number of information gathering techniques available especially relevant to defining the requirements of Web applications (Standing, 2001:553). ICDM uses
group communication techniques, and believe that this can speed up the definition of the logical requirements for a Web application. The two group communication techniques used in ICDM are:

- Brainstorming
- Group Requirements Sessions (GRS)

Brainstorming is used to define a few alternative ways of using Internet commerce in the business. Group requirement sessions obtain the detailed requirements within a relatively fast time frame with involvement from customers, suppliers, and internal staff. ICDM aims to develop prototypes, using these to help with development of the definition of the requirements. Attention should be given to detailed information requirements of transactions and marketing systems. The prototype will however be used to a greater degree in the design phase of development.

3.2.2.7 Functional requirements framework
Web applications fall into a number of categories and need detailed definitions of their requirements. ICDM does not give deep insight into these requirements. However, the analyst needs to use the analysis techniques (3.2.2.6) to make sure that the business objectives are being met.

3.2.2.8 Physical architecture framework
Techniques used for defining the requirements of an Internet project depend on the type of system and its functionality. There are three fundamental types of Web systems (Standing, 2002:157):

1. Document publishing systems
2. Basic interactive systems
3. Complex transaction systems

Web projects do not always require complex transaction systems when there might be an indication of possible transformations within the organization. Useful information with some simple database interactivity has the capability of making major impressions on the business.
3.2.2.9 Design phase
ICDM does not discuss the design phase in depth. The design phase involves designing the network infrastructure, developing the web site, and developing security controls. Web site design should consider (Standing, 2001:554):
- Desired Image
- Usability
- Promotion
- Evaluation with customers

3.2.2.10 Implementation and evolution phases
The implementation of the Web site relates to meta-development strategies (3.2.2.3). Web applications constantly evolve and rarely have a well defined project completion. There is however components that remains reasonably stable. One example might be the Web site's implemented transaction modules. The continual evolution of the Web site should be monitored by the organization's Web management team. It is the team's task to oversee the implementation of the Web strategy and changes in strategic direction. They should also create a policy highlighting 'who' can add to the Web site, Web site content, and its guidelines (Standing, 2002:554).

3.3 Web Engineering: A new discipline for the development of Web-based systems

3.3.1 Introduction
Web Engineering was developed by Murugesan (1998). The following discussion of Web Engineering was based on information collected from Murugesan (1998) and Murugesan et al (2002). Web Engineering, an emerging new discipline, advocates a process and a systematic approach to development of high quality Internet- and Web-based systems. Web engineering is believed to be more complex than traditional Information System engineering. It raises many new issues such as presentation issues, user profiling, navigation support etc.

According to Murugesan et al (2000) the broad and objective definition of Web engineering is as follows:

"Web Engineering is the establishment and use of sound scientific, engineering and management principles and disciplined and systematic approaches to the..."
successful development, deployment and maintenance of high quality Web-based systems and applications."

Web Engineering combines inputs from various diverse areas (figure 3.8), most attention given to engineering principles, but other fields include social sciences and graphic design.

Web Engineering can not be seen as a single activity or task, as it deals with most of the aspects related to Web-based system development. It addresses phases for realization, development and. It also continues with regular maintenance and performance evaluations.

Web Engineering includes some of the following activities (Murugesan et al, 2000:5):

- Requirements specification and analysis
- Web-based system development methodologies and techniques
- Integration with legacy systems
- Migrating legacy system to Web environments
- Web-based real time application development
- Testing, verify and validate
- User-centric development
- End-user application development
It is believed that Web Engineering principles and approaches could be advantageous in bringing potential chaos in Web-based systems development under control, minimize risks, and enhance maintainability and quality. The Web engineering logo (figure 3.9) depicts the philosophy, objective and goals promoted by Web engineering: a framework and methodology for Web-based systems development that supports creativity and flexibility, and still retains and respects the characteristics and the features of the Web medium.

![Web Engineering Logo](image)

Figure 3.9: Web Engineering Logo (Murugesan et al. 2000)

The outer square symbolizes a broad framework and guidelines for Web-based systems development. The spider web inside the square represents the World Wide Web. The spider web implies that there is room for creativity, flexibility, and adoption to specific applications.

### 3.3.2 Framework for development of successful E-Business systems

e-MINDER (Electronic Commerce Leveraging Network for Developing European Regions) proposes the use of a framework which aims at improving the quality of e-commerce systems accommodating small to medium organizations. This can be done by addressing the issue of complexity of the application to be produced and associated time and man-effort resulting by this complexity. The framework is based on a derivative of Business Process Re-engineering that assesses various aspects related to the client's organization. Aspects include the identification of critical business and organizational factors that contribute to the complexity of the Web-based system. These factors are combined with any significant issues arising from the application domain, and the requirements describing the desired quality.

All this information is then transferred into measures of complexity that drive the main decision of whether the development process of the system should follow a certain version of the Web engineering process. The versions available include a short:
process cycle (SPC), or a full, long process cycle (LPC). The terms “SPC” and “LPC” refer to the time aspect of the process, and the required effort to be put by the human resources of the developer.

The e-MINDER framework outlines three primary categories of critical factors that could affect the complexity level of a Web-based system and the length of the development process:

- Business and organizational issues.
- Application domain aspects.
- Quality requirements.

3.3.2.1 Business and Organizational issues
According to e-MINDER, Business process Re-engineering (BPR) can help organizations attain better productivity by dramatically changing existing business processes. BPR can be identified by five critical issues:

i. BPR consists of radical or at least significant change.
ii. BPR’s unit of analysis is the business process.
iii. BPR’s purpose is to achieve dramatic performance improvements.
iv. Information Technology is a critical enabler of BPR.
v. Organizational changes are a critical enabler of BPR.

Performing successful BPR requires seven phases (Covert, 1997):

a) Begin organizational change,
b) Build the reengineering organization,
c) Identify BPR opportunities,
d) Understand the existing process,
e) Reengineer the process,
f) Blueprint the new business system,
g) Perform the transformation.

The proposed framework utilizes the BPR in an attempt to assess the current state of the organization, to explain the need for change, and to illustrate the desired position it aspires to capture via the E-commerce system. This process can reveal critical factors about the client organization and the business performed that orient the answer to the question whether the organization is ready to move to e-commerce and give indications as regards the level of the development effort needed for building a new or modifying an existing system.
The Business Assessment (BA) is the first set of activities the software analyst must adopt, placing emphasis on three key areas: a) Business goals, b) Market aspects and c) Human resources. Business goals are the primary determinants of the type of the system to be developed. Achieving these goals means examining the market and deciding where and when the organization wishes to bid the market and investigating whether the human resources of the organization are adequately trained and willing to support the new processes. Analytically:

a) Business Goals
The software analyst investigates the business perspective of the client organization and transforms it to the type of services needed for the e-commerce system to promote this perspective. These services are then assessed in terms of feasibility and complexity. Certain types of services may not be feasible due to unrealistic technical requirements, budget limitations or time constraints.

b) Market Aspects
The analyst and the management investigate whether the demands of the marketplace are shifting. If this is the case then new and innovative services may need to be developed, something that leads to demanding a higher level of development effort. If not, then the analyst and the organization explore ways for developing better e-business applications and processes than those offered by competitors who have already made significant advancements in products or services, or in online business. Analyzing market competitiveness will provide a clear view as to the upper time bounds of the development process and will also reveal the type of services that must be offered to win the competition or at least be equally competitive.

c) Human Resources
Management and personnel may also hide critical success factors in developing an e-commerce application. Therefore, the analyst takes into consideration several human issues which can be decomposed into the following categories and corresponding elicitation activities:

1. Management Awareness and Involvement – Explore management awareness about e-commerce in general and true involvement in the project. Also, define the level of management support and the areas in which it will be involved.
2. Personnel Training – Identify the key personnel that will be involved in the reengineering process, their level of education, the areas that need to be further trained and the available time bounds.

3. Human, Social and Organizational (HSO) Factors – Investigate HSO factors that may affect the system development or slow down the re-engineering process, such as communication and collaboration issues, personnel’s willingness to change their working procedures, user expectations of the system, whether users feel threatened by the system in terms of current posts or job loss, etc.

A Suggested action list with questions supplied by e-MINDER can help in assessing the business and organization issues of the SME as follow:

1. Business Goals
   a. Investigate the business perspective of the SME.
   b. Transform the SMEs perspective to the type of new services needed for an e-commerce system.
   c. Assess new services feasibility in terms of technical requirements, budget and time limitations.
   d. Assess new services complexity in terms of technical requirements, budget and time limitations.

2. Market Aspects
   a. Investigate the demand for these services.
   b. Define Competition
   c. Analyze market competitiveness for these services.

3. Human Resources Issues
   a. Define management’s level of awareness and involvement.
   b. Identify the personnel’s involvement and its level of knowledge.
   c. Identify significant HSO factors within the SME which are important and they might affect the systems deployment.

3.3.2.2 Application Domain

To assist in estimating the level of complexity and development effort required, various types of Web applications can be divided into the following most common application categories (Dart, 1999):

1. Informational – Read only content, data with navigation and links to related data sources and other relevant Web sites.
2. Download – Information or data available that can be downloaded by a user.
3. Customizable – Depending on the customers needs, the content can be customized accordingly.
4. Interaction – Communication can be used among users via chat rooms, e-mail, or instant messaging.
5. User input – Inputs can be received from users thru the use of online questionnaires or forms.
6. Transaction oriented – When the Web is used for online handling of products, or exchanging of services.
7. Service oriented – The application provides an online service to the customer, for example the estimation of an insurance premium.
8. Portal – A beginning point that channels the user to other Web applications outside the domain of the current portal application.
9. Database access – Querying a database and retrieving necessary information.
10. Data warehousing – Querying a collection of large databases and retrieving information as required by the customer or in this case an organization.

3.3.2.3 Quality Requirements

The complexity of a Web application can be viewed in terms of quality requirements such as usability, functionality, reliability, efficiency and maintainability (Olsina et al., 1999):

1. Usability
A Web application must be implemented in such a way that it can be easily understood in terms of its functioning and actions by either an intermediate or beginner classed user.

Important design factors that the Web engineers in charge of development cannot afford to miss include:

- Understandability
- Friendliness
- Operation ease
- Playfulness
- Ethics
- Learn ability
The design of an appealing easy understandable user-interface, consistency and easy usage are all attributes of these easy-to-learn systems with rapid learning curves.

2. Functionality
The Web application must include all the necessary features to accomplish the required task(s) or to meet the relevant requirements. Important aspects to look at that need to be investigated to ensure that the Web application will perform as expected include: Accuracy, suitability, compliance, interoperability and security. The Web application must have searching and retrieving capabilities, navigation and browsing features and application domain-related features as highlighted in paragraph 1.2.2 (Olsina et al., 1999).

3. System Reliability
e-MINDER states that when producing a reliable Web system, the designers need to understand issues such as: fault tolerance, crash frequency, and recoverability. The system needs to maintain a specified level of performance in case of software faults with the minimum crashes or down-time possible. The Web system needs to consistently produce the same results, and meet or even exceed users’ expectations. The Web application must have correct link recognition to other sites or pages, user input validation and recovery mechanisms in the case of a system crash.

4. Efficiency
A Web-based system’s goal is usually to increase productivity, decrease costs, or a combination of these two. Customers expect the system to run in an efficient manner in order to support their goals. System’s response-time performance, as well as page and graphics generation speed, must be high enough to satisfy the user demands. Fast access to information must be examined also throughout the systems life to ensure that user requirements are continuously met on one hand, and that the system remains competitive and useful on another.

5. Maintainability
Some crucial features related to maintaining a Web application is its:
- Analyzability
- Changeability
- Stability

39
Testability

The primary target here is to collect data that will assist designers to conceive the overall system in its best architectural and modular form, from a future maintenance point of view. With the rapid technological changes especially in the area of Web engineering, as well as the rigorous user requirements for continuous Web site updates, easy system modifications and time dependant enhancements, both in content and in the way this content is presented, are also critical success factors for the development and improvement of a Web-based system.

3.4 Extreme Programming (XP)

3.4.1 Introduction

Extreme Programming is believed to be a lightweight, low risk and flexible approach of software development based on underlying principles of simplicity, communication, feedback and courage. Extreme Programming, or XP, was developed by Kent Beck (2000), who wrote the pilot book on the subject. XP can be described as a lightweight, low-ceremony, high discipline methodology. Although XP is fairly new to the Web-based systems development genre, Wallace et al. (2002) believes that XP can be adapted and applied to the Web-based system development process.

The rest of this methodology discussion uses information collected from Beck (2000). XP is a collection of rules and practices each of which supports several others, and are supported by several others in turn. XP works by bringing the whole development team together by the use of simple practices. There is the option of enough feedback to enable the team to conclude where they are in terms of the project scope, and the chance to tune the practices to their own unique situation.

XP can be distinguished from other methodologies by (Beck, 2000:7):

- XP’s early, concrete, and continuing feedback from short cycles.
- XP has an incremental planning approach. This quickly comes together with the overall plan that is expected to evolve through the life of the project.
- Business needs change, and XP’s flexible schedule allows it to adopt to the business’s implementation of functionality.
XP relies on automated tests written by customers and programmers. This monitors the progress of development, and to identify defects early.

XP relies on good communication channels thru verbal communication, tests, and source code to communicate the structure of the system and its intent.

The followed evolutionary design process that lasts only as long as the system lasts.

Its reliance on the close collaboration of programmers with ordinary skills, and their practices that work with both their short, and long-term interests.

3.4.1.1 The XP Team

XP is designed to be used with projects that can be built by a team ranging from two to ten programmers, with schedules not constrained by the existing computing environment, and where a reasonable job of executing tests can be done in a fraction of a day. The team must also include a business representative called the "customer", or "client".

This client provides the requirements, sets the priorities, and steers the project. It would be advantageous if the client is an end user, whom has knowledge of the domain, and expected requirements. The client can also make use of Analysts to help set forth these requirements. The team could also include testers, who can with the client help to create Customer acceptance tests. A Manager might be appointed to provide resources, handle communication within the organization and coordinate activities.

XP believes that these roles are not the exclusive property of any one individual, and anybody on the team can contribute. XP also highlights that the best teams have no specialists, but general contributors with special skills.

3.4.2 The Basics

XP addresses the basic problem found in its development, namely deciding on how the basic activities of its system development should be approached. These activities include coding, testing, listening, and designing.

XP has a set of guiding values and principles to aid in choosing the best strategies for each activity. Each will now be discussed in turn:
Extreme Programming embraces four values (Beck, 2000):

1. Communication
2. Simplicity
3. Feedback
4. Courage

1. Communication – Numerous projects have had setbacks, and even failure due to the lack of communication between the project team. For example a programmer that did not tell the team of a critical change in design, or a manager not asking the right questions and misreporting the project’s progress. XP aims to keep communication active, by employing practices that can not be completed without communicating. These practices will be discussed shortly and include unit testing, pair programming, and task estimation. XP employs a coach whose job it is to monitor communication, and if necessary, take necessary steps to close this gap.

2. Simplicity - XP follows the approach that it is better to do a simple task today, and change it at minimal cost later if something is wrong, than to invest in a huge complicated project which might never be used. XP believes that the simpler the system is the less communication is required, which in turn leads to more complete communication. This narrowed down simple system also requires a smaller scope, and less unnecessary resources.

3. Feedback – Feedback works in different time scales. Firstly in minutes and days. The programmers have minute-by-minute feedback on the state of their system. If any changes are required, the programmers immediately know about the request and can address the problem. Feedback can also work on the scale of weeks and months. Customers help unit testers write functional tests, which on a weekly basis tests the current state of the system. If in any of the previous two case a change needs to be done, feedback helps in identifying the problem early.

4. Courage - XP’s design strategy can be compared to a hill-climbing algorithm. It starts off with a simple design, and then gets more complex as the tip approaches. The problem is that when the algorithm approaches local optima, a little change won’t have any visible influence. Here a large change is required, and courage is required to take action in a case where the output is unsure. XP believes that courage combined with communication, simplicity and concrete feedback can be very valuable (Beck, 2000).
These four values give XP its criteria for a successful solution. XP acknowledges that these values are too vague in helping in choosing the right practices to use. XP therefore sub-divide these values into concrete principles that can be used. Here are the fundamental principles (Beck, 2000):

I. Rapid Feedback – Psychology teaches that the time between an activity and the feedback received is critical to learning. XP aims to get feedback, interpret it, and put the new found knowledge back into the system as quickly as possible.

II. Assume Simplicity – Treat every problem as if it can be solved with ridiculous simplicity. This is a difficult principle for programmers, who are taught to plan for the future and design with reuse in mind. XP follows the approach that a member solves the days job first, and trust on the ability of the member to add complexity later on in the future where needed.

III. Incremental change – XP believes that big changes made all at once don’t work. It applies incremental changes in many ways. The design, plan and team all change only a little at a time. Even the adoption of XP is to be taken in small leaps.

IV. Embracing change – A good strategy is to preserve most options available, while actually solving the applicable problem.

V. Quality Work – Of the other development variables available, quality is the least negotiable. The only possible values are “right” and “excellent”. As some systems could have people’s lives at stake, this variable should not be neglected.

3.4.3 Applying Practices

The next stage in XP is to structure the four activities, and decide how they will take place in development. XP uses a set of guiding values (discussed earlier) and principles to guide the team as they choose strategies for each of the four activities.

All the practices, except for testing, need to be kept in balance. Figure 3.10 summarizes the practices. The pieces alone are simple, but the richness comes from the interaction of these principles.
Each of the principles will now be discussed as defined by Beck (2000):

1. The Planning Process

Planning in XP addresses two questions in Software development. These two questions focus on guiding the project. They try to predict what will be accomplished by the due date, and determine what to do next on the project. There are two planning steps available in XP to address these questions:

   - **Release planning** is where the client gives the functional requirements needed from the system to the programmers, and they then estimate the difficulty involved. The client then lays out a plan for the project with known knowledge of the importance of certain functions, and associated cost estimates.

   - **Iteration planning** helps in giving the team direction every couple of weeks. The XP team builds software in two-week "iterations", and is expected to deliver useful running software at the end of each iteration.

These planning steps are quite simple, but provide very good information and excellent steering control in the hands of the client. The client has clear visibility on the amount of progress that has been made, and can cancel if it is deemed insufficient. The client can also use this visibility to decide what will be done next on the project without any hidden scope issues that might have emerged.
2. Small Releases

Every release on the system should be as small as possible, and contain the most valuable system requirements. XP teams should be careful not to release half a feature just to make the release cycle shorter, but rather a release that makes sense as a whole. XP plans no more than a month or two at a time, and believes it is much better to update frequently on a very short cycle.

3. Metaphor

XP teams develop a common vision of how the system functions, which they call the "metaphor". The metaphor in XP replaces much of what other people call "architecture". This along with a common set of names helps everyone on the team to understand how the system works, and where to find the functionality that they require. By asking for a metaphor, the client and team are likely to get an architecture that is easy to communicate and elaborate.

4. Simple design

A system built with XP should be the simplest program that meets the current requirements. The design should be suited for the current functionality of the system, and should have no wasted motions. There are steps in release- and iteration planning, plus teams participate in quick design sessions and design revisions throughout refactoring. This is done through the entire course of the project. In an incremental, iterative process like XP, good design is essential, and this practice requires a lot of attention.

5. Testing

XP teams focus on the validation of the system at all times. Programmers use unit tests to gain confidence in the operation of the program, and create software that fulfills the requirements reflected in the tests. The client also creates acceptance tests, which aid in determining if the features they need are provided. By frequent testing, the XP team can only improve on the system. By building up on pieces that have already passed the tests, it is believed that the system can only grow.

6. Refactoring

When implementing a program feature, the programmer should always look at if there is a way of changing the existing program, to make the addition of a new...
feature simpler. After adding the feature, the programmer can look at how to make the program simpler, while still running all of the tests. This is called refactoring. The goal is for the XP team to improve the design of the system throughout the entire development cycle. This is done by keeping the software clean, without duplication and with high communication.

7. Pair Programming
All production software in XP is built by two programmers, working together at one machine. According to XP, this practice ensures that all the production code is reviewed by at least one other programmer, and results in better design, testing, and coding. If one programmer has responsibility for a task in an unfamiliar area, they might ask someone with recent relevant experience to pair with them. More often, anyone on the team can be chosen as a partner.
Pair Programming has been used in many experiments to show that it produces better software at the same and even lower costs, than with programmers working alone.

8. Collective Ownership
All the code in the system belongs to all the programmers. Anybody who sees an opportunity to add value to any portion of the code is required to do so at any time. In XP, everyone takes responsibility for the whole of the system. If a pair of programmers is working and they see an opportunity to improve the code, they should go ahead and do it if it makes the system better. This lets the team work at full speed, because when something needs to be changed, it can be done so without delay.

Collective ownership can be a problem when a programmer works blindly on a piece of code that the programmer doesn’t understand. XP tries to avoid this kind of problem with the use of programmer test and pair programming discussed earlier.

9. Continuous Integration
XP teams integrate and build the software system multiple times per day. This aims to keep all the programmers on the same page, and helps in very rapid progress. One simple way to do this is to have a computer dedicated to integration alone. When a pair of programmers has a piece of code to integrate,
they load the current release and their changes. They then change code where needed, until they past all the tests.

10. 40-Hour Week

XP believes that tired team members only make more mistakes. XP teams should not work excessive overtime, and keep themselves fresh, healthy and effective. Overtime is a symptom of a problem in the project. XP has a simple rule, only one week of overtime is allowed, because XP believes that a project that requires more overtime than this, can't be solved by adding more hours.

11. On-site Customer

An XP project requires a customer or client to be available to answer questions, determine requirements and set small scale priorities. The on-site customer will have the disadvantage of being physically separated from other customers, but will most likely have time to do their usual work. The advantage of being there is that communication improves, and there is also less documentation required, which sometimes costs quite a lot of money.

12. Coding Standards

XP teams have to follow a common coding standard. This helps in the sense that all the code in the system looks like it was written by a single individual, and supports the collective ownership practice. The standard should emphasize communication, and be adopted voluntarily by the team.

3.4.4 Adopting XP

Don Wells describes the simple solution on how to adopt XP (Marchesi et al., 2002):

1. Pick the worst problem

2. Solve it with the four activities in mind, and apply it with knowledge found from XP’s values, principles and practices.

3. When the problem is reduced start at step one again

The easiest place to start is at testing and the planning game. Many projects have quality problems, or with a contradiction between business and development. XP finds many advantages in this approach, as the team learns only one practice at a time, and gain a lot of thorough knowledge by doing it so. Another advantage is
that because the team always addresses the biggest problem, they have a lot of motivation for the required change, and get immediate visible feedback for their efforts. XP also claims that with adopting each practice, it will be customized for the current situation.

XP concludes that if the team isn’t faced with a problem, it shouldn’t be considered in solving it with the XP approach.

3.5 The Relationship management Methodology (RMM)

3.5.1 Introduction

The relationship Management Methodology (RMM) for hypermedia design was introduced by Isakowitz et al (1995). Since then, this methodology has evolved in a number of ways to accommodate the rapid growth in demand for hypermedia applications on the Web. This discussion is based on information gathered from Isakowitz et al (1995). Hypermedia application design is believed to differ from other software design because it involves navigation, user-interface and information processing issues. Other differences might include:

- Hypermedia projects may include people with completely different skill sets. For example, authors, content designers, programmers and even artists.
- The design of the Hypermedia applications involves capturing and organizing a complex domain, while also making it clear and usable by the user.
- Multimedia aspects involved in Hypermedia design raise numerous difficulties, and the need for prototyping and testing is very high.
The rest of this chapter uses information sourced from Isakowitz et al (1995) to discuss this methodology. The Relationship Management Data Model (RMDM) forms the cornerstone of the RMM methodology. The model is strongly based on the most used way of modeling software applications currently, the Entity-Relationship model. RMDM includes elements for representing information domain concepts such as entities and their relationships. Also included are navigational elements, for example links. In figure 3.11, application design is described via an RMDM diagram.

With E-R design, a study of the relevant entities and relationships of the application is conducted. These elements form the basis of the hypermedia application and show up as nodes or links. When navigational design is done, relevant relationships are identified and made available to be used in navigation. Because units can contain a lot of information, RMM tries to group the units into slices called M-slices. The modeling primitives of RMDM are shown in figure 3.12. In the top part of the figure there are domain primitives. These model information about the relevant application domain, and include entities and their attributes.
The attributes are grouped into slices as seen in the middle of the figure. The entities might consist of a large number of attributes, and it could be impractical to show all of these of the instance at once, hence the groupings. Navigation is supported in RMDM by the six access primitives shown in the bottom of the figure: unidirectional and bi-directional links, for access between slices of an entity. Grouping which can be seen as a construct supporting access or an index and act like a table of contents to a list of entities. Conditional indexing can be qualified by a predicate namely a guided tour that can be defined as a linear path through a collection of items. And a conditional indexed guided tour, which represents a guided tour qualified by a predicate.

The links establish an association between the original data and the destination data components. The link is what effectively allows navigation through data components, and represents the main characteristics of hypertext and hypermedia. The possibility of different link types establishes semantic information that allows hypermedia...
application systems to efficiently manage data or data modeling. This prevents the task of being the sole responsibility of designers or users. The available link types can be used throughout nearly all the steps of model-based approaches to hypermedia design. RMM believes that presently static, persistent, and explicit links are found in most hypermedia systems. RMM uses links to initiate certain processes such as e-mail, video, audio, or even a file download. Destination influence establishes what of the operational services are going to be supported by the link.

A framework is presented to categorize link types. This leads to the design and construction of richer hypermedia applications. Links are the kernel of model-based approaches to hypermedia design, and their design can benefit by the usage of these links. Information and operational elements also influence link creation, so these issues should be taken into consideration when designing link efforts.

3.5.2 The RMM methodology

RMM is a methodology used for hypermedia application design based on the relationship management data model. The Methodology consists of seven steps which are displayed in figure 3.13 in the form of S1 to S7.
The 7 steps are described next:

1. E-R Design

The first step is to represent the domain of information on the application by using an Entity-Relationship diagram. RMM uses this E-R diagram, as it is familiar to many system analysts, and is well documented. The diagram can also model information dependencies in numerous application domains. This step of RMM represents a study of the relevant entities and relationships of the application's domain.

These entities and relationships will form the roots of the hypermedia application, and many will be visible in the form of nodes and links in a hypermedia web system. In RMDM there are associative relationships that appear in the E-R
diagrams. These relationships represent associations between entity instances. The relationship can consist of one-one, one-to-many and many-to-many arities.

2. Slice Design

The next step is unique to hypermedia applications. It determines how the information in the entities will be presented to the end-user, and how the user will access it. The approach is to split an entity into meaningful slices and organizing these slices into a hypertext network. The information can be divided into meaningful units that could be presented as separate but interrelated wholes (Isakowitz et al., 1995). The organization of entities into slices is called the slice design phase, and can be viewed in a slice diagram. In this diagram each entity has a “slice head”, which is used to anchor links coming into the entity.

The entity diagram also models navigation between the different slices with the use of uni-directional and bi-directional links. The links connecting slices are called structural links. Structural links differ from associative relationships in the sense that they connect information pieces within the same entity instance. Associative relationships on the other hand interconnect different entity instances, which most likely belong to different entity classes.

RMM believes it is important to differentiate among these kinds of connections. When the associative link is used, the information context changes, for example a University that is sub divided into its different schools. However, when a structured link is used, the context of information remains the same within the entity.

3. Navigational Design

The navigational design steps aids in designing the paths that will enable hypertext navigation. Every associative relationship that appears in the enriched E-R diagram is analyzed. If an associative relationship is required for navigation, it is replaced by one or more RMDM access structures. Because RMM is mostly used on domains that are updated frequently, all the navigational paths are specified in generic terms. This approach allows no hard coded links between instances of entities. A better option would be to specify the links by referring them to properties of entities and their relationships. This is made possible through three RMDM navigational elements: conditional slices, conditional guided tours and conditional indexed guided tours (Isakowitz et al., 1995).
Navigational design starts by designing the navigation between the entities, which are based on associative relationships. The name of a relationship is used as a condition in the access structures, and indicates the instances of entities that are interconnected. The next part is to group the items of interest, and design a high level access structure. These groupings then act as hierarchical menu-like access to different entities. The access structure enters by default, an entity via its head slice, but the designer can change this entry if required. This is done by tagging the access structure with the name of the newly selected target slice.

At the end of the navigational design step, the enhanced E-R diagram has been transformed into an RMDM diagram, which describes all access structures to be implemented into the system.

Steps 4 to 6 can be performed simultaneously from the RMDM diagram.

4. Conversion Protocols design
   This step uses a set of conversion rules to transform each element of the RMDM diagram into an object in the chosen platform. Conversion protocol design mostly performed manually by programmers, but an automatic conversion programme is also available. An example could be a list-box, which can be used to represent an index.

5. User Interface design
   User interface design involves designing presentations for every RMDM object. This includes information content of slices identified in the second step, and access structures to linked information from the third. Also included are the locations of orientation and navigation components, as well as the layout of nodes. This includes button layouts, appearance of nodes, indices and location of navigational aids.

   User interface design could include either low fidelity prototyping using a pencil and paper, or high fidelity prototyping using computerized tools. This phase is similar to constructing an interaction state diagram but less formal.

6. Run-time behavior design
   In the sixth step, the programs that control how the application generates and retrieves information are designed, and how it interacts with the user. The team
should decide how link traversal, history, backtracking and navigational mechanisms are to be implemented. Developers should also consider the volatility, and the size of the domain, and if link end-points are to be built during the development phase, or dynamically computed at runtime. Another aspect includes designing the algorithms and implementation mechanisms for hypermedia navigation. This includes indexed navigation, history tracking and browsing.

The design team should identify the parts of information content that need to be generated dynamically, which constitutes the link base. They should also identify the programs that are invoked by the links.

7. Construction and testing

The final step implements the design image obtained through the first six steps. It involves the construction of a physical database from the logical design done in step one. The database is then populated with domain data, and implemented with the mechanisms designed in the previous steps.

Special attention should be given to testing. It is important to test all the navigational paths in hypermedia applications and ensure that it satisfies functional, navigational and usability requirements. For Hypermedia applications this includes testing the links to ensure that the proper underlying parameters are generated.

3.5.3 Conclusion

The RMM methodology is believed to be most suited to applications that have a regular structure. It is especially relevant where there is a frequent need to update information and keep the system current. Applications that fit this description include: product catalogs, manuals, interfaces to database management systems and electronic commerce gateways. The RMM methodology aims at being a basis for the design and development of robust hypermedia applications (Isakowitz et al., 1995:14).
3.6 Summary

In this chapter five methodologies used for Web-based systems development were discussed. Firstly WISDM was discussed. WISDM utilizes a framework during the development of Web-based systems. After this ICDM was discussed, this aims at business-to-consumer application development, as well as provides a guiding framework for developers.

Web Engineering was conversed which can be seen as an emerging new discipline that advocates a process and systematic approach to development of Web-based systems. The fourth methodology discussed was Extreme Programming which follows development based on underlying principles of simplicity, communication, feedback and courage. The last methodology discussed was RMM that places focus on hypermedia application design. RMM believes this differs from other software design because it involves navigation, user-interface and information processing aspects.
Chapter 4

Methodology Evaluation

In this section the author will compare the selection of methodologies according to the framework suggested by Avison and Fitzgerald (2003). The selection of Web-based systems development methodologies will also be theoretically evaluated for their suitability to develop Web-based systems. There are mainly two reasons for comparing methodologies. On one side a methodology is analyzed academically. This analysis will aid in understanding the nature of the different methodologies better and help in the classification of their different approaches. The other side is more for a practical reason, which can help in choosing a methodology for an organization to follow in their particular application.

4.1 Methodology Comparison

Avison and Fitzgerald (2003) identified a framework for comparing methodologies. This framework can be used as a basis for comparing different methodologies.

The Avison and Fitzgerald (2003) framework consists of 7 elements:
1. Philosophy
2. Model
3. Techniques and tools
4. Scope
5. Outputs
6. Practice
7. Product

1. Philosophy: A philosophy distinguishes, more than any other criterion, a ‘methodology’ from a ‘method’. The choice of the areas covered, the systems, data or people orientation, the bias or otherwise toward a purely IT solution, and other aspects are made on the basis of the philosophy of the methodology. In most methodologies the philosophy is implicit, but in some cases can be explicit. Many feature analyses have neglected or skipped this aspect because methodology authors seldom put enough emphasis on their philosophy.

The four factors of a philosophy are discussed next:
a) Paradigm
There are two relevant paradigms. The science paradigm, which has characterized most of the hard scientific developments of recent times, and secondly the systems paradigm which is characterized by a holistic approach. Science breaks things down into smaller parts for examination and explanation, and copes with complexity through reductionism. This breaking down does not interfere with the system from which it is a part of. Checkland (1981) argues that human activity systems do not have these characteristics, and that they have emergent properties, meaning i.e. that the whole is more important than all the small parts together. This led to the development of the systems approach. Both science and systems paradigms are closely related to the concepts of hard on soft systems thinking.

b) Objectives
A fairly obvious clue to the philosophy of a methodology can be found in the stated objective or objectives. There exists a difference between some methodologies. Some are only interested in aspects that are 'computerizable', while others take a wider view and direct their attention to achieving solutions or improvements. This difference is important in the sense that it determines the boundaries of the area of concern.

c) Domain
The domain of a methodology reflects on the situations that it addresses. Early methodologies saw it as their task to overcome only a particular or sometimes narrow problem. The disadvantage of this approach was that the solution of those problems on an ad hoc basis at different time intervals, could lead to a few different physical systems being in operation at the same time. This led to a number of methodologies adopting a different philosophy, taking a much wider view of their starting point, and not just trying to solve only particular problems.

d) Target
The last aspect is the applicability of the methodology. Some methodologies are targeted at particular types of problems, while others are said to be general purpose. (i.e. environmental, or type or size of organization)

2. Model: The second element of the Avison and Fitzgerald (2003) framework concerns an analysis of the model that the methodology adheres to. A model which normally represents a view of something is used in this case to represent the
methodology's view of the world. It is an abstraction and representation of the factors important to the organization. The model can act as a means of communication, it can be a way of capturing the essence of a problem or design, and can be a representation which provides insight into the applicable problem area.

Categories of models:
1. Verbal
2. Analytical or mathematical
3. Iconic, pictorial, or schematic
4. Simulation

In the information systems field, the models are usually of the third type. The reason for the dominance of iconic, pictorial, or schematic models is the perceived importance of using the models as a means of communication mainly between users and analysts. The model also ensures that all the information needed is captured at appropriate stages in development.

3. Techniques and Tools: A key element in the Avison and Fitzgerald (2003) framework is the identification of the techniques and tools used in a methodology. A technique is a way of doing a certain activity in the applicable phases within the methodology. Some methodologies may use many of these activities in their approach. In table 4.1 a brief summary can be seen of some of the techniques available. In the first column of the table the technique type and in the second column are examples of the type of technique.
<table>
<thead>
<tr>
<th>Type</th>
<th>Example</th>
</tr>
</thead>
</table>
| Holistic techniques        | - Rich pictures  
|                            | - Root definitions  
|                            | - Conceptual models  
|                            | - Cognitive mapping |
| Data techniques             | - Entity Modeling  
|                            | - Normalization |
| Process techniques          | - Data flow diagramming  
|                            | - Decision trees  
|                            | - Decision tables  
|                            | - Structured English  
|                            | - Structured diagrams  
|                            | - Matrices  
|                            | - Action diagrams  
|                            | - Entity life cycle |
| Object orientated techniques| - Object orientation  
|                            | - UML (Unified Modeling Language) |
| Project management techniques| - Estimation techniques  
|                            | - PERT charts  
|                            | - Gantt charts |
| Organizational techniques  | - Lateral Thinking  
|                            | - Critical success factors  
|                            | - Scenario planning  
|                            | - Future analysis  
|                            | - SWOT  
|                            | - Case-based reasoning  
|                            | - Risk analysis |
| People techniques           | - Stakeholder analysis  
|                            | - Joint application development |

Table 4.1: Techniques used in System development methodologies (Avison and Fitzgerald, 2003)

These techniques can use certain tools to aid in the process. These tools normally represent some of the artifacts used in information systems development. Organizations can acquire most of these tools commercially, for example Groupware, Dreamweaver®, Visio, Microsoft Project®, etc.
4. **Scope:** The Scope of a methodology is an indication of the stages of the systems development life cycle of systems development which the methodology covers. Avison and Fitzgerald (2003) identify nine stages in development:

a) **Strategy:** Is used to indicate if a methodology addresses any aspects which relate to an organization-wide context, and that deals with overall information system strategy, planning, and purpose, rather than just that of a particular system area of concern.

b) **Feasibility:** Defined as the economic, social, and technical evaluation of the system under consideration.

c) **Analysis:** This stage includes user requirement analysis.

d) **Logical design:** This stage is covered by all methodologies except SSM (Soft Systems Methodology) and PI (Process Innovation).

e) **Physical design:** This stage is concerned with the physical design of the system, and is not relevant to the more ethical methodologies which are less explicit.

f) **Programming:** The programming stage can be associated with the physical development of the system.

g) **Testing:** Includes the planning as well as the testing of systems, programs, and procedures.

h) **Implementation:** In implementation we include planning and implementation of technical, social, and organizational aspects.

i) **Evaluation, maintenance:** This stage concerns the measurement and evaluation of the implemented system. Maintenance is only covered if it is specifically addressed in terms of tasks within the methodology.

Scope is not the only element in analyzing a methodology. Depending on the purpose of the comparison, and the methodologies involved, other dimensions may be more appropriate. The problem with using these stages is that there are methodologies that do not follow a systems development life cycle.

5. **Outputs:** This element in the Avison and Fitzgerald (2003) framework concerns the outputs from the methodology. It is important to know what the methodology is producing in terms of deliverables at each stage, and the final deliverable itself. This can vary from an analysis specification to a working implemented system.
6. Practice: The next element of the Avison and Fitzgerald (2003) framework is termed practice and is measured according to
   a) Methodology background in terms of academical or commercial approach
   b) User base
   c) Participants in a methodology

The practice should also include an assessment of difficulties and problems encountered. It is also important to assess any differences that might appear between the practice and the theory of the methodology.

7. Product: The last element of the Avison and Fitzgerald (2003) framework is the product of the methodology. This concerns the product in terms of what the buyers actually get for their money. This might be a software program, or documentation, consultancy, and so on.

4.2 Methodology Comparison according to proposed framework

Using the Avison and Fitzgerald (2003) framework, the methodologies discussed in chapter 3 were compared in a tabular form. The results are presented in table 4.2 which follows.

The seven elements are grouped on the left of the table, and the methodologies in the columns that follow to the right of the table. A short description is given in each cell to highlight the differences.
<table>
<thead>
<tr>
<th>Short Description</th>
<th>WSDM</th>
<th>ICDM</th>
<th>Web-Engineering</th>
<th>Extreme Programming</th>
<th>RMM</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Philosophy</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Level of usage of a paradigm: objectives, questions and target</td>
<td>Follows a soft system approach that covers both analysis and design of system. The Multiview framework provides a basis for constructing a situation-specific methodology.</td>
<td>ICDM has a paradigm, aimed at business-to-consumer application development that is intensely social in nature. ICDM also provides a guiding framework for developers.</td>
<td>Web Engineering follows a scientific approach that advocates a process and systematic approach to development of Web-based systems.</td>
<td>Extreme Programming is a flexible approach of software development based on underlying principles of simplicity, communication, feedback and courage.</td>
<td>RMM is a data driven methodology most suited to applications that have a regular structure. The methodology aims at being a basis for the design and development of robust hypermedia applications.</td>
</tr>
<tr>
<td><strong>Model</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Specified model that the methodology is based on or adheres to</td>
<td>This methodology's model is categorized as iconic, pictorial, or schematic. The methodology also adheres to an underlying Multiview framework.</td>
<td>This methodology's model is categorized as iconic, pictorial, or schematic. The ICDM methodology uses a framework for developing strategies, and the evolutionary development of Web-based systems.</td>
<td>This methodology's model is categorized as iconic, pictorial, or schematic. Methodology centered on Extreme programming principles, rules and practices.</td>
<td>This methodology's model is categorized as iconic, pictorial, or schematic. Methodology based on the Relationship Management Data Model.</td>
<td></td>
</tr>
<tr>
<td><strong>Techniques and tools</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| Identification of the most visible techniques and tools used in the methodology | - UML Tools  
- Conceptual Modeling  
- Use Case Diagram  
- Class Diagram  
- Interaction Diagram  
- Sequence Diagram  
- State Transition Diagram  
- Activity Diagrams  
- Object-Orientated Analysis and Design | - XML Tools  
- CAEM Process Re-engineering  
- Value Chain Analysis  
- SWOT Analysis  
- User Involvement Framework | - XML Tools  
- Unit Testing  
- Acceptance Testing  
- Pair Programming  
- Capability Maturity Model  
- Index Cards | - RMM-Case Tool  
- RMMCM Modeling Diagram  
- ER Diagram  
- Site Design  
- Navigational Design  
- Conversion Protocol Design  
- User Interface Design  
- Run-Time behavior Design  
- Testing |     |
<table>
<thead>
<tr>
<th>Scope</th>
<th>Outputs</th>
<th>Practice</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stages of the SDM life cycle of a Web-based system which the methodology covers</td>
<td>Deliverables produced at each stage of the methodology</td>
<td>Methodology background in terms of an academic or commercial approach</td>
</tr>
<tr>
<td>Framework provides a basis for WISDM’s situation specific methodology which then utilizes techniques and tools to solve specific problem situations. WISDM covers the whole development cycle.</td>
<td>After analysis WISDM covers the following: Work design, Technical design and Human computer interface design. The end product is a complete Web-based system.</td>
<td>Methodology used in a commercial domain, utilizes some of the principles found in academic solutions.</td>
</tr>
<tr>
<td>ICDM covers the whole development process and is a business analytics methodology as well as a systems development methodology.</td>
<td>Initially creates strategies for business and implements these strategies as part of developing the commerce system. The end product is a complete Web-based system.</td>
<td>Methodology used in a commercial domain, utilizes some of the principles found in academic solutions.</td>
</tr>
<tr>
<td>WebE covers the development, deployment and maintenance of complete Web-based systems.</td>
<td>WebE combines various areas to supply a full life cycle covering aspects from analysis to implementing system. The end product is a complete Web-based system.</td>
<td>Methodology used in a commercial domain, utilizes some of the principles found in academic solutions.</td>
</tr>
<tr>
<td>XP uses a simple form of planning and tracking to decide what should be done next and predict when the project will be done. The team produces the software in a series of small fully-integrated releases delivering a complete system at the end of the development process.</td>
<td>XP teams build software in two-week “iterations”, delivering running useful software at the end of each iteration. The end product is a complete Web-based system.</td>
<td>Methodology used in a commercial domain, utilizes some of the principles found in academic solutions.</td>
</tr>
<tr>
<td>RMM methodology focuses only on the design, development and construction phases.</td>
<td>EIR Diagram, ER+ Diagram, RM Diagram that is used in design and construction. RMM aims at being the basis for the development process that delivers a complete Web-based system.</td>
<td>Methodology used in a commercial domain, utilizes some of the principles found in academic solutions.</td>
</tr>
<tr>
<td>Product</td>
<td>End goal of the methodology</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-----------------------------</td>
<td></td>
</tr>
<tr>
<td>WSWM</td>
<td>analyzes the problem, and designs a solution around this to complete or enhance a current Web-system through the use of the Multiview 2 framework.</td>
<td></td>
</tr>
<tr>
<td>ICDM</td>
<td>provides a framework for developing Web-based systems. It is a loose fitting framework for developing strategies and the evolutionary development of Web-based systems.</td>
<td></td>
</tr>
<tr>
<td>WebE</td>
<td>deals with most of the aspects related to Web-based system development. It addresses phases for realization, development and also continues with regular maintenance and performance evaluations.</td>
<td></td>
</tr>
<tr>
<td>XP</td>
<td>The XP team releases running, tested software, delivering business as required by the Customer, at every iteration. There are design steps in release planning and iteration planning, plus teams engage in quick design sessions and design revisions through refactoring, through the course of the entire project life.</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.2: Comparison of Methodologies according to Framework

A working application developed using RMM with a RMDM diagram that highlights all the entities and links in the system.
All the software development methodologies highlighted in table 4.2 focus on the same end goal, which is to supply a complete Web-based system. It is interesting to note that the biggest difference between the methodologies is the philosophical approach followed by each one. WISDM follows a soft systems approach, with ICDM also following a more social paradigm. WebE is focused mainly on a scientific approach and XP on a more flexible approach. The last methodology, RMM, follows a data driven approach.

It is also interesting to note that there are a lot of similarities between the different methodologies. All the methodologies have a model that can be categorized in the iconic, pictorial, or schematic group. Nearly all the methodologies also aim to deliver a complete Web-based system, with the exception of RMM, which focuses only on the design, development and construction phases.

Although all the methodologies deliver a Web-based system, there is no mention of any methodology supplying CASE tools to maintain and support the Web-based system after implementation.

4.3 Theoretical evaluation of selected system development methodologies for their suitability to develop Web-based systems

The goal of this theoretical evaluation is to determine if the methodologies developed for Web-based systems development are suitable for developing these kinds of systems. Web-based systems have certain characteristics that require more attention, and assist in this differentiation from normal software systems. The characteristics discussed in chapter 2.3 earlier in this study, will be used to do a theoretical evaluation of the methodologies discussed in chapter 3 for their suitability to develop Web-based systems. Some of the characteristics which imply duplication were removed for this evaluation. The theoretical evaluation is displayed in table 4.3.

Although there are a lot of methodologies available for Web-based system development, these methodologies are not used as much in practice as their developers would like. By using the theoretical evaluation, this study hopes to identify a few possible reasons why these system development methodologies are not used to such a great extent.
The characteristics are summarized according to the rows, and the different methodologies grouped according to the columns. In the table, the characteristics of the methodologies that are believed to be suitable or applicable are highlighted by the grayed out cells. In every cell there is also a short description of why the characteristic is believed to be suitable or not to the defined methodology.
<table>
<thead>
<tr>
<th>Description</th>
<th>Extreme Programming</th>
<th>WISDM</th>
<th>RMM</th>
<th>ICDM</th>
<th>Web-Engineering</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Multimedia Environment</strong></td>
<td>Web-based systems offer a multimedia environment which can represent a combination of media format, including text, image, graphic, sound, voice, and animation.</td>
<td>XP focuses more on coding of the system, and less attention on the GUI.</td>
<td>WISDM focuses an engineering approach that relies on rationality. It also includes a HCI activity in the development process</td>
<td>RMM utilizes a user interface design step that is included.</td>
<td>Business and external focus is more important to the ICDM methodology.</td>
</tr>
<tr>
<td><strong>Integration of various information bases</strong></td>
<td>Supports systems integration of various kinds of information and conflict information bases.</td>
<td>Methodology constructs Web-based system with databases integration as needed.</td>
<td>WISDM utilizes two main components that include combining the front and back end of the Web-based system.</td>
<td>RMM creates a set of all the information sources allowing integration.</td>
<td>Web Engineering includes a graphic design phase.</td>
</tr>
<tr>
<td><strong>Interactive communication support</strong></td>
<td>Web-based systems support interactive communication in general. This high level of interaction gives users dynamic control of information.</td>
<td>The XP team and client design a metaphor, which is a common vision of how the system functions. This following might be limiting the Web-based system only to their vision.</td>
<td>WISDM utilizes HCI - Human Computer Interface in its design approach.</td>
<td>RMM includes system behavior design.</td>
<td>ICDM’s meta development strategy gives user input power and can understand their needs better.</td>
</tr>
<tr>
<td><strong>Support of networks for information access</strong></td>
<td>Supports networks to access information. Web-based systems create truly interactive networks with information exchanges between users and servers.</td>
<td>XP’s system requirements specify usability through available connections.</td>
<td>WISDM utilizes both the front and back end through various channels.</td>
<td>Different networks can interact between the links that RMM defines.</td>
<td>Web Engineering highlights the importance of usability in its methodology.</td>
</tr>
<tr>
<td>Provision of a cross-platform environment</td>
<td>A cross-platform highlights that web systems can be accessed independently on various computer operating systems.</td>
<td>The high emphasis placed on coding in XP can be software specific, and not independent.</td>
<td>The Web-based system could run on all platforms, but there is no guarantee for the database platform which it utilizes.</td>
<td>The Web-based system could run on all platforms, but there is no guarantee for the database platform which it utilizes.</td>
<td></td>
</tr>
<tr>
<td>Adherence to time pressure</td>
<td>For Web-based systems it is important to take advantage of any short-term gain to feed to more long-term advantages.</td>
<td>Extreme programming only approaches short term to replace it with evolutionary design concept over the long term.</td>
<td>The speed of development is dependent on the WISDM framework which can initially lengthen time utilization.</td>
<td>The speed of development highlighted with ICDM's fast prototyping.</td>
<td></td>
</tr>
<tr>
<td>Requirements available</td>
<td>Does the methodology allow for proper requirements analysis?</td>
<td>XP codes and adjusts as required. There is a low initial requirements definition.</td>
<td>WISDM's methods within the framework will gather requirements when highlighted.</td>
<td>The RRM doesn't cover requirement analysis in depth.</td>
<td></td>
</tr>
<tr>
<td>Prototyping importance</td>
<td>Web based systems rely on a software prototype as requirements definition.</td>
<td>XP starts to code and adjusts the Web-based system from the first step in the methodology. There is a low initial requirements definition.</td>
<td>WISDM uses both the business strategy and SSM to gather information for analysis.</td>
<td>ICDM utilizes a logical functional requirement analysis phase.</td>
<td></td>
</tr>
<tr>
<td>Release time</td>
<td>Web based systems are released early and often and suggest early release and frequent re-release.</td>
<td>Extreme programming is focused on writing quick releases, which adds complexity in an evolutionary approach.</td>
<td>WISDM is a more in-depth methodology that adapts its framework to each new requirement.</td>
<td>ICDM places high focus on speed of development.</td>
<td></td>
</tr>
</tbody>
</table>

Web Engineering supports SPL (Short Process Lifecycle).
<table>
<thead>
<tr>
<th>Parallel development support</th>
<th>Extreme Programming allows programmers to code different activities simultaneously.</th>
<th>WSDM utilizes the multiview framework, following problem solutions as they arise. There is no mention of addressing problems in parallel.</th>
<th>RMM follows a certain cycle, for example the design phase follows the modeling phase.</th>
<th>ICDM attempts to address the issues related to emphasizing a business focus, external focus, and speed of change that consumer applications development requires. There is no mention of the possibility of addressing these issues in parallel.</th>
<th>Web Engineering allows for strict roles and parallel development possibilities.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coding requirements</td>
<td>In the Web environment, developers must be able to code problems out quickly minimizing down time, and be able to make changes quickly if required.</td>
<td>Extreme Programming allows the option to &quot;code on the fly.&quot;</td>
<td>Coding will have to be structured into the WSDM multiview framework.</td>
<td>Through taking advantage of channeling through dynamic links, codes can be modified, and the link referral updated to the new piece of code.</td>
<td>ICDM places emphasis on addressing problems through the SWOT analysis. This reduces the chance of just starting on coding.</td>
</tr>
<tr>
<td>Staff requirements</td>
<td>Web-based systems are completed under time pressure and typically in small teams where all members need to pull their weight. Key staff can make or break a project.</td>
<td>Extreme Programming enhances team work by forcing developers to work in pairs.</td>
<td>WSDM encourages using multiple perspectives in its methodology approach.</td>
<td>RMM focuses mainly on Hypermedia design and in this regard is not applicable.</td>
<td>ICDM includes the possibility to use different teams in its development process.</td>
</tr>
</tbody>
</table>

Table 4.3: Theoretical evaluation of system development methodologies

70
It is important to take note that Table 4.3 is only a theoretical evaluation, and that one must be careful not to make generalized conclusions. Only five Web-based systems development methodologies were used in the evaluation, although many more are available. Another aspect to remember is that not all organizations focus the same amount of attention on all the characteristics. Certain characteristics could be very important to one organization, and not even be required by another. Furthermore, the list of characteristics may not be complete, and some system development methodologies might have certain characteristics which are only applicable to their approach.

It is interesting to note from the theoretical evaluation that the two characteristics that all the methodologies adhere to, both have to do with information integration. The two characteristics are firstly the integration of various information bases and secondly the support of networks for information access. Both these characteristics place focus on two aspects that are most applicable to Web-based systems. That is to be able to gather information from networks and Web platforms, and to integrate all this through a central Web-based system.

A lot of emphasis is placed by organizations on the adherence to time pressure by these development processes. This is one characteristic which less than half the methodologies take into consideration. Web Engineering and ICDM are the only methodologies that adhere to this requirement.

This theoretical evaluation also suggests that Extreme Programming and WISDM only adhere to around 50% of the characteristics. Web Engineering addressed most of the characteristics by complying with nearly 90% of them. Based on the information derived from this theoretical evaluation, it seems that the Web Engineering methodology satisfies most of the characteristics associated with Web-based systems.
4.4 Summary

In this chapter the methodologies discussed in chapter 3 were compared. In table 4.2 the methodologies were compared based on the framework suggested by Avison and Fitzgerald (2003). Although this table highlights some of the differences quite clearly, it does not give some kind of indication of which methodology might be more suitable for Web-based systems development than the others discussed. Therefore, in table 4.2 a theoretical evaluation of these methodologies was performed to determine their suitability for Web-based systems development.
Chapter 5

Research Method

In this chapter the qualitative research method followed as well as the techniques and methods used for data collection and data analysis will be discussed. The qualitative research design for this study was based on the underlying main research question:

*How (and why) are systems development methodologies used in Web-based application development in South Africa?*

This underlying question guided the researcher during the research design. This study will follow the qualitative research approach. One of the main goals is to determine "how" system development methodologies are used in practice to develop Web-based systems, and to what extent it is used. In order to gain a deep understanding of the above, it is necessary to use a qualitative approach. Qualitative research can help in drawing a conclusion of how the different parties use or don't use methodologies for development, and to articulate and understand it.

5.1. Research approach

It is important to note that there are broadly two approaches one can follow while doing research, namely a quantitative and a qualitative research approach. In earlier times, natural sciences such as biology, physics, geology, chemistry etc. were researched with extra detail given into the investigation of things that could be observed and measured in some way. These measurements and observations could be made objectively and repeated by any other researcher. This process is known as "quantitative" research (Hancock, 1998:5).

Later on in time researchers working in the social sciences were interested in studying human behavior and the social world inhabited by human beings. They found it increasingly difficult to explain human behavior in simple measurable terms. These measurements show how many people behave in a certain way but they do not adequately answer the question of "Why do people behave in this way?" Research which attempts to increase the understanding of why things are the way they are in our social world, and why people act in the ways they do is "qualitative" research. Qualitative research is concerned with the social aspects of our world.
Hoepfl (1997) combined several sources to identify the following characteristics associated with qualitative research:

1. Qualitative research attempts to observe, describe and interpret the environment in its natural state. Because qualitative research has been described as naturalistic, researchers try to minimize or control the effects that they might have on their informants.

2. The researcher fills the role of a "tool" to collect data. The qualitative researcher suspends, or sets aside, his or her own beliefs, perspectives, and predispositions.

3. Qualitative researchers predominantly use inductive data analysis.

4. Qualitative research reports are illustrative, incorporating expressive language and the "presence of voice in the text" (Eisner, 1991:36).

5. Qualitative researchers try to understand people from their own frame of reference. By doing this they try to understand how these people see things from their perspective.

6. Qualitative researchers seek uniqueness in every case that they study.

7. Qualitative research follows an emergent design. Researchers focus on new processes, as well as on the current outcomes or products from that research.

8. Qualitative researchers emphasize validity in their research. Qualitative methods are designed to ensure a close fit between the data and what people actually say and do.

Each of the various features of qualitative research may be viewed as either a strength or a weakness. This mainly depends on the original purpose of the research. For example, one criticism levied at qualitative research is that the results of the study may not be generalizable to a larger part of the population because the sample group was too small and not randomly chosen. But this all depends on the type and aim of the research. The research may have sought insight into a specific subgroup different from the original population (Hancock, 1998). Such a subgroup can for instance be a group of scientist trying to create hydrogen power. In such a study focusing on a small group is more applicable, and would be more appropriate than addressing a large portion of the population.
5.2 Research methods

There are a few different types of qualitative research methods available. Hancock (1998) identified the following four major qualitative research types:

1. Phenomenology

Phenomenology means literally the study of phenomena. It can be seen as a way of describing something that exists or forms part of the world we live in. Phenomena may be events, situations, experiences or concepts. Phenomenological research begins with the acknowledgement that there is a gap in our understanding. Clarification or illumination of this gap will be of benefit. Phenomenological research will not always lead to a definitive explanation, but can guide to raise awareness and increase insight.

2. Ethnography

The term means "portrait of a people" and it is a methodology for descriptive studies of cultures and people. The cultural parameter is that people under investigation have something in common. For example, religious or tribal beliefs, even people sharing a particular experience. Analysis of data concluded from this approach adopts an "emic" approach. This means that the researcher attempts to interpret data from the population under study. The results from this study are conveyed as if they were being expressed by the subjects themselves. Use of the applicable local language and terminology to describe the phenomena is not uncommon.

3. Grounded theory

This research approach aims to the development of new theory through the collection and analysis of data on the phenomenon. These explanations that emerge are new knowledge and are used to develop new theories about the phenomenon. A key feature of grounded theory is the simultaneous collection and analysis of data using a process known as constant comparative analysis. In this process, data are transcribed and examined for content immediately following data collection. Ideas emerging from the analysis are included in data collection when the researcher next enters the field. Basically new theory begins its conception as the researcher recognizes new ideas and themes emerging from what people have said or from events which have been observed.
4. Case Study

Case study research is one of those research approaches which can take a qualitative or quantitative standpoint. Case study research is used to describe an entity that forms a single unit such as a person, institution, or an organization. Some of these research studies describe a series of cases.

Case study research ranges in complexity. Research can range from an illustrative description of an occurrence, to a more complex analysis of a social situation over a specified period of time. The most complex is the extended case study. This approach traces events involving the same actors over a period of time, enabling the analysis to reflect changes and adjustments. Case study research is a highly versatile research method that employs any and all methods of data collection from testing to interviewing.

For this research, the multiple-case study design method was used. A case study was created for each of the organizations identified later on in this chapter. Focus was placed on specific “how” and “why” questions used to gather information. The same questions were used at each organization, to enable this research to analyze the data individually and to do cross-case analysis on the gathered data. These questions will be discussed in 5.3.

Yin (1994) stated that a study's propositions are derived from these questions, and that they are helpful in keeping the study fixed on its goals. Yin (1984) also presented three conditions for the design of case studies:

1. The type of research questions raised. In this research focus was placed on specific “how” and “why” questions.
2. The extent of control a researcher has over the actual behavioral events. The researcher had no control over the events that took place, and this is an important characteristic of case study research.
3. The extent of focus on contemporary events. This research study examined events that are contemporary, although certain historic information was also included.

The result of using case study research is that the researcher might gain a sharpened understanding of why the organizations use system development methodologies for Web-based systems development. It can also highlight certain points which need to be analyzed and addressed in this research.
Another aspect to consider was to decide on the target for data capturing. Organizations targeted in this study should have Web-based systems development noted as one of its specialty areas. Organizations targeted included:

- Large Organizations focused mainly on public services including Banks, Government affiliates or departments.
- Large Organizations that supply services to the enterprise market for example Development houses or Consulting firms.
- Smaller Organizations in the private sector for example an Organization only developing Web artifacts.

By targeting these three organizational types, this research can focus on three different areas that develop Web-based systems in the South African market, and provides a richer understanding if focus is placed on different areas. Letters were sent to candidates who specialize in developing Web-based systems, explaining the general goal of this research. After the organizations were identified, a specific project within the organization was chosen for this research.

Three organizations that surpassed these criteria were selected:

- Standard Bank South-Africa
- Fourier Approach
- Working Webs cc

By referring to the classes of Web-based systems identified by Kaiser (2000) that was discussed earlier in chapter 2, it can be noted that each of the three chosen projects fall into a different class. Standard Bank's Internet banking system can be classified as a self service Web-based system. Fourier Approach developed an information management system for the government to use in its project management duties, and Working Webs designed a Web-based system that can be categorized as falling under the business platform.

The three organizations will each be discussed individually next.
5.2.1 Standard Bank South-Africa

Standard Bank is one of South-Africa's biggest banking groups. The group has a wide global base, representing 17 African Countries, and over 21 countries outside Africa. The Organization has three sub categories which includes Personal & Business Banking, Corporate & Investment Banking and Investment Management & Life Insurance.

![Image of the Namibian Internet Banking system]

The Web-based system that was studied is the Internet banking system for Standard Bank in Namibia. An example of main page of this Web-based system can be seen in Figure 5.1. The system is the same as the current legacy system used in South-Africa, except that certain functionality constraints had to be adhered to. Certain banking functions which weren't required in Namibia had to have a workaround option, and technical requirements originating from a smaller infrastructure had to be met. Standard Bank describes the function of the system as:

"Standard Bank Internet Banking is a convenient way of doing most of your banking from your home, office or anywhere else you have access to the Internet. This online service allows you to perform your banking transactions through the Internet on accounts linked to your Auto Bank and E Plan cards. You may also link up to nine accounts to a card. If your time is precious, our Internet banking service is for you. It gives you anytime, anywhere access to a host of functions, including a way of paying..."
your monthly accounts. You’ll find that it will give you more time to spend on other aspects of your life.”

5.2.2 Working Webs

Working Webs have been specializing in Web site design and development since 1995. They offer services that include consultation and training, and also offer packages for full dynamic content web sites.

A Web-based system called Formgrids was analyzed. According to Working Webs Formgrids was developed to enable the user to do their homework as easily possible in studying race horses. History and specified horse’s performance can be viewed in easy to understand grids, aiding in helping a betting person to track a horse’s odds. Formgrids currently contains information from South Africa, Mauritius, Zimbabwe, United Kingdom, United States, Hong Kong and Canada. The data includes results and race entries for each of these countries. A screen shot can be seen in Figure 5.2 bellow.

Feedback from an initial run of the system was positive, with quotes from various areas stating:

- “Excellent concept. I am amazed at the speed of the form studying.” Tex Lerena, Chairman of the South Africa Jockey Association.
• "I have used horse racing sites all over the world, but this site is in a league of its own." Andrew Bonn, Horseracing TV Presenter.

• "Excellent, this is so easy to use to study form. Very good, I like it very much." Kevin Shea, Master Jockey.

5.2.3 Fourier Approach Pty (Ltd.)

Fourier Approach is a privately owned company operating in the information technology and information systems arena. The company was founded in March 1999 with the merger of various consulting firms. Fourier Approach renders business and systems analyst consulting services, information system acquisition, process definition and management.

The Web-based system discussed is called the PROMPT system. It can be summarized as a program management system, with programs defined as a set of projects grouped together to manage the strategic goals of the company.

Although an example of the system can not be shown due to license and competition constraints, it is noted that the system is utilized by the local government and is mostly used by project managers whom have to maintain and run huge amounts of projects simultaneously. The goal of the PROMPT system is to enable the Project manager to align the strategic drives within the business with the detailed projects currently taking place. The PROMPT system handles tasks like aiding in defining the project requirements, setting up a project scope and milestones, and maintaining the completed project after finalization.

Fourier Approach has been steadily growing in the past few years, with a client base that includes:
5.3 Data collection

Qualitative approaches to data collection usually involve direct interaction with individuals on a one to one basis or in a group setting. Data collection methods are time consuming and consequently data is collected from smaller numbers of people than would usually be the case in a quantitative approach. The benefits of this approach include richness of data and deeper insight into the phenomena under study (Hancock, 1998:14).

There are certain methods to use when collecting qualitative data. Some of the methods are described next:

I) Observation

Observation presents a series of methods designed to minimize or eliminate the researcher's effect on the people and setting being studied. This is what is meant by "non reactive research". Observations may include descriptions of the participants, descriptions of the physical settings, and accounts of particular events and activities (Bogdan & Biklen, 1998).

Hancock (1998, 17) identified some techniques for collecting data through observation:

- **Written descriptions.** The researcher can record observations of people, a situation or an environment by making notes of what has been observed.
- **Video recording.** This frees the observer from the task of making notes at the time and allows events to be reviewed time after time.
- **Photographs, films and artifacts.** Photographs are a good way of collecting observable data of phenomena which can be captured in a single shot or series of shots. Research filming is a powerful tool for inquiry into past events. Film has the unique ability to capture visible phenomena seemingly objectively, yet always from the perspective of the filmmaker. Research film methodology requires the documentation of the
time, place, and subject of the filming, as well as the photographer’s intent and interest.

- **Documentation**: A wide range of written materials can produce qualitative information. People’s own diaries, letters, pictures, records, calendars, and memorabilia can be used to guide interviews without imposing a structure on informants. When researchers have a body of direct experiences to build on, they can be somewhat more directive and aggressive in their initial questioning.

The strong point of unobtrusive methods can also be its weakness. Since researchers do not interact with people, they not only eliminate reactive effects, but fail to learn how people see and experience their world (Rice and Ezzy, 1999).

II) Interviews

Interviews can be divided into two groups namely group and one-to-one interviews. In group interviews the researcher can bring together groups of people to talk about their everyday tasks in free-flowing, open ended discussions.

Group interviews can be used when (Hancock, 1998:15):

- Limited resources prevent more than a small number of interviews being undertaken.
- It is possible to identify a number of individuals who share a common factor and it is desirable to collect the views of several people within that population sub group.
- Group interaction among participants has the potential for greater insight to be developed.

The disadvantage with group interviews is that the researcher probably never gains the depth of understanding that comes with one-to-one interviews. For this research one-to-one interviews were used. The following advantages of one-to-one interviews have been identified (SIECUS, 1996):

- Allows greater depth than questionnaires
- Data is deeper, richer and has more context
- Interviewer can clarify questions that might be misinterpreted
- Is a good method for working with low literacy respondents
- Higher response and completion rates
- Allows for observation of nonverbal gestures
Qualitative interviews should be fairly informal and aim to follow the semi-structured or unstructured interview method. Interviewees should feel as if they are participating in a conversation or discussion rather than in a formal question and answer situation. However, achieving this informal style is dependent on careful planning on skill in conducting the interview (Hancock, 1998).

III) Official records and Public documents

There are, for all practical purposes, an unlimited number of official and public documents, records, and materials available as sources of data. These include organizational documents, newspaper articles, agency records, government reports, court transcripts, and a host of other materials.

Although there are a lot of methods available, this research used the one-to-one semi-structured interview method for data acquisition. According to Case (1990) semi-structured interviews are conducted with a fairly open framework which allow for focused, conversational, two-way communication. Semi-structured interviews encourage two-way communication, and those being interviewed can ask questions to the interviewer. It is believed that often the information obtained from semi-structured interviews will provide not just answers, but the reasons for the answers (Case, 1990).

Interviews were informal with a list of questions that were used to guide the interview. Discussions could freely move to other topics, but the set questions helped in keeping the interview focused on answering questions related to the research goal, which as described earlier is to determine "how" system development methodologies are used in practice to develop Web-based systems, and to what extent it is used.

It also focuses on the prevailing characteristics associated with this. In addition it aims to understand the context in which these Web-based systems are developed. Questions were developed with the main research goal in mind, focus then shifted to the mini research questions which aimed to get a better understanding of how systems are developed in organizations. After this the questions were weighed against four chosen question groups which they had to satisfy in order to satisfy the research goal set forth by this research. The four question groups were divided accordingly.
A. What does the Software system entail?
B. What environment is the system set in?
C. Who were the participants of system development?
D. How was the system developed?

The process can be seen in figure 5.3.

To understand the context in which Systems are developed

To understand how systems are developed:
- SDM usage / extent of
- SDM effectiveness

With these groupings in mind, two different lists of questions were developed to guide the interviews. One consisting of questions more aimed at the business and managerial side, and the other aimed at the developers and programmers. With this in mind, this research designed different questionnaires for different team members in the project. This aided in inferring a deeper understanding from the different ranked participants in the project.

None of the interviewees were aware that this study focused on system development methodologies. A discussion on Web-based system in general was held, as not to influence the interviewee’s thoughts, or place focus on methodologies to influence the outcome of this study.

The list of questions that was used to guide the interview with the team / project leader consisted of the following questions:
1. Please shortly describe the Web-based system according to the following:
   a) Description of the system
   b) Goals of the system
   c) The needs of the system
   d) Description of the physical system itself
2. What requirements were set for the system, and did the system meet these requirements?
3. How did the extra requirements set forth by this Web-based system affect your development?
4. What was the level of importance set on the system?
5. How big is the current Web-based system? Its reach, lay-out and user count?
6. What different platforms does the system run on?
7. Who and how do they handle support and maintenance of the Web-based system?
8. How many people and different groups does the development team consist of?
9. How many people are required for administration of this system?
10. Please describe a typical user of your system.
11. Did users play a role in the development of your system?
12. Shortly describe the development process (Term SDM not mentioned)
13. Does everybody doing development follow this process?
14. Do you always use this process, or does it sometimes get adapted to meet other requirements?
15. Why do you follow this process?
16. Are there any extra set guidelines or rules for development?
17. How long did the development process take?
18. Were there a lot of changes required after testing and implementation?
19. Do you think your system was a success?
20. Did the development process have any influence on the success (outcome) of the project?

The next list of questions that was used to guide the interview with the Developers / Programmers included:

1. What is your primary role in systems development?
2. Did you receive any additional training before starting with the development of applications / systems?
3. What procedures were followed before you start with the actual development and programming?
4. Did you make use of any processes in development?
5. Did you make use of any techniques?
6. When developing a system, what steps or processes do you normally follow?
7. Do you always follow this process or make changes when necessary?
8. Are there any other set guidelines or rules when developing an application or system?
9. How long does a typical application take to develop? Do you think this development process can be quickened, and if yes, how?
10. After the system was implemented, were there a lot of changes required?
11. Could these changes have been avoided?
12. Do you think the applicable application/system was a success?

When an interviewer conducts an interview, there is the option of taking either notes, or recording the verbal interview. Recording the interview is preferred for a number of reasons. First being that the interview can flow uninterrupted. The interviewer does not have to cause between questions to write down responses, or secondly, be distracted by recalling what has already been said. Because the interviewer does not have to scribe what he/she thinks is important, it can also help to minimize bias. As in the case with the recording, everything is noted, and did the interviewer not only note and write down what they wanted to hear. For this study the interviews were all recorded with the consent of the organization, and the interviewee. Anonymity was promised to all the participants.

Interviewees consisted of team members who ranged from programmers and developers to team leaders. Interviews were arranged with the chosen organizations without specifying the goal of this research to the interviewees, as to not influence the interview outcome and minimize bias. Interviews were conducted on each organization's own premises. Meeting rooms were allocated for the different interviews, with the interviews ranging from an hour for the project leader, to 45 minutes for the developers and programmers. The Standard Bank case study consisted of interviews with a project leader and two developers individually. Both Fourier Approach and Working Webs case studies consisted of interviews with the project leader, as well as one of the developers individually.
After the interviews were completed, the next phase is to transcribe all the information. Transcribing can be seen as a protocol for producing a written document of the interview. It is important to transcribe the whole interview, as also to minimize bias, and give a more complete overall picture. Transcribing is a time consuming process with an estimated ratio of time required for transcribing interviews is about one to five (Hancock, 1998:14). This ratio implies that a one hour interview would require five hours to transcribe completely. All the interviews were transcribed on hard and soft copies.

5.4 Data Analysis

To analyze data in a research project, it is required of the researcher to group and summarize the huge amount of data that has been collected. This grouping of data needs to be in such a way, that the end result communicates the results relevant to the desired outcome. Qualitative research is mostly centered on describing a phenomenon, and to express it in a means that can be understood.

Firstly all the raw data were collected through the semi-structured interview method discussed earlier. The next step was to arrange the raw data in to related groups, according to topics or related themes. After this the titles or headings were formed for the newly founded data groupings. These headings will form the stepping stone for data analysis that follows shortly.

For this research content analysis was used. It uses certain procedures to make valid inferences from text (Weber, 1990). The content can be analyzed on two levels, namely the descriptive account of data, and the latter being interpretative. Content analysis involves coding and classifying data. The procedure involves a series of steps developed by Hancock (1998) that are listed as follow:

1. Take a copy of the transcript or collected data and read through it. Make a brief note of any interesting information that might have been noticed.
2. Look through the margin notes from step 1 and make a list of the different types of information that have been found.
3. A list of items excerpted from the text or data has been made. Read through the list of data items and categorize each item in way that describes what it is about.
4. Looking at the categories identified from the transcripts or collected data, consider whether some of the categories may be linked in some way. Also list the categories as either major or minor categories depending on size and importance.

5. Look through the list of minor and major categories of data and compare and contrast the various categories. As a "bigger picture" develops, some items of data might be seen as more "fitting" in an alternative category. Sometimes an item seems to belong in two categories. If so, list it under both.

6. Move on to the next transcript or collection of data and repeat the process from step 1 - 5. Eventually there will be no need for any new categories, and all the existing data be accommodated in the available categories.

7. Collect all the extracts from the transcripts and collected data that have been put into one directory because they appear to bear some relationship to each other. Examine each of the extracts in turn.

8. When the entire relevant transcript and other data have been sorted in minor and major categories, look again at the data contained in each category. If any data still needs to be moved from one category to the next, do it now.

9. Once all the categories are sorted, look at the range of categories to see whether two or more categories seem to fit together. If so, they may form a major theme in the research.

10. Go back to the original copies of the transcripts and collected data. Look at any text that might have not been included because it might have not been relevant at the time. The end results are themes, major categories and minor categories clearly sorted.

The process requires the researcher to move and group the data as needed. When new data or facts emerge changes might be required. After changes are not required for the groupings anymore, the end result will be categories that are precise in reflecting the content of the data. A researcher which sorts and organizes their data must know what it is that forms the data in the context of the research as a whole.

Following the same principle as the steps developed by Hancock (1998), the collected data was imported into AtlasTi®, which is a qualitative data analysis application. AtlasTi® facilitates many activities used in text analysis and
interpretation. It uses tools to handle, extract, compare and utilize meaningful segments of large data amounts in a user friendly way.

Firstly all the transcribed text was imported into the Atlas Ti programme. Key fields were identified and grouped according to a list of codes. As in the case with Hancock (1998) that suggests that categories should be made, Atlas Ti gives a researcher the option to create different code families. The second step sees all the codes being assigned to their relevant “Family”. This step helps in defining the code better, and to confirm whether all the texts are grouped correctly in relevant codes and groups, or families. This process was repeated for the three different organizations, linking all the cases together. By following this approach certain codes associated within the different organizations could be highlighted.

By being able to successfully compare the different organizations, cross-case analysis can be used to derive conclusions in chapter 6. Cross-case analysis can aid in identifying shared goals, procedures and results among the selected body of research, focusing on system development methodology usage where applicable.

Palmquist (1997) highlights that this analysis method aids in keeping the researcher from making premature conclusions, by making sure that the researcher looks at the data in different ways. Cross-case analysis separates the data by type across all the cases investigated. When a pattern from one data type is supported by the evidence from another case, the finding is stronger. When evidence produces conflicts, a deeper investigation of the differences is needed to identify the source of the conflict (Palmquist, 1997).

In this research cross-case analysis is applied to the data to analyze it across all the cases in order to identify similarities and contradictions between the different systems development methods used by the organizations for Web-based systems development. By identifying these similarities and contradiction, this research seeks to provide further understanding into issues concerning the development of Web-based systems by the relevant organizations.
5.5 Confirmation of findings

This research will infer certain conclusions based on the analysis of the data as discussed in 5.4. Propositions of the conclusions will be generated to substantiate the main goal of this research, namely to determine whether system development methodologies are used in practice to develop Web-based systems in South Africa, and to what extend. The aim is to build up a “weight of evidence” to support the conclusions (Seaman 1999:569). According to Lincoln and Guba (1985) the applicable research should stand against certain questions which the trustworthiness of the project can be evaluated against. These questions include:

1. How credible are the particular findings of the study? By what criteria can we judge them?
2. How transferable and applicable are these findings to another setting or group of people?
3. How can we be reasonably sure that the findings would be replicated if the study were conducted with the same participants in the same context?
4. How can we be sure that the findings are reflective of the subjects and the inquiry itself rather than a creation of the researcher’s biases or prejudices?

Lincoln and Guba (1985) also refer to these questions as establishing the “truth value” of the study, its applicability, consistency, and neutrality.

Seaman (1999:569) proposes six approaches that can help in supporting the generated findings:

1. The first approach places importance on the researcher to ensure the validity of the methods or procedures used to gather information and data.
2. Triangulation is another important approach for confirming the generated theory. The intention is for the researcher to collect data through different methods, and by gathering different types of evidence be able to support a proposition.
3. Anomalies in the data can be used to explain, shape or even support a conclusion.
4. Another approach is to search for evidence that might contradict the research findings. If the research can proof these contradictions false, it would also have a good groundwork to stand on.
5. Replication is believed to be an expensive tool to confirm findings. It suggests a replication of the study to compare conclusions. For qualitative research it would only be necessary to perpetuate the conclusions set forth in the theory.

6. The last approach is called member checking. This approach presents the feedback of the conclusions to the subjects who provided the data in the first place.

For this study the member checking approach was utilized. The conclusions derived in chapter 6 were sent back to the three organizations. Each project leader reviewed the conclusions and gave their feedback, and with the final draft, approval on the findings.

5.6 Summary

In this chapter the research approach used for this study has been identified as the qualitative research approach. Three organizations were identified who have passed the criteria required by this study. At each organization the semi-structured interview method was used to gather data, along with the one-to-one interview method discussed earlier in this chapter.

For data analysis, content analysis was performed as this enabled the use of cross-case analysis after the data gathering stage. Cross-case analysis was used to derive similarities and contradictions between the different case studies. By identifying these similarities and contradiction, this research seeks to provide further understanding into issues concerning the development of Web-based systems by the relevant organizations.

The outcome from the data analysis phase will aid in the discussions that follow in chapter 6, and lead to the main aim of getting a deeper understanding of how organizations develop their Web-based systems, and what systems development methodologies they use to accomplish this.
Chapter 6

Results

In this chapter the results of the study will be presented. After the data were gathered, all interviews were transcribed. The Standard Bank case study consisted of interviews with a project leader and two developers. At both Fourier Approach and Working Webs interviews were conducted with the project leader, as well as one of the developers. For each organization content analysis was performed using Atlas Ti², in order to obtain a deep understanding of how they develop Web-based systems.

In the first part of this chapter the findings for each organization will be discussed. Cross-case analysis was performed to derive similarities and contradictions between the different organizations. In the second part of the chapter the results obtained from the cross-case analysis will be discussed. The results will be structured as follows: the context in which Web-based systems are being developed in the organizations will be discussed, followed by a description of the systems development methodologies used in the organization, and the influence of the systems development methodology on the quality of the system.

6.1 Findings for Standard bank

6.1.1 The Team

The Standard Bank team consists of members who are fully employed, as well as contractors. Team members fill various roles acting as developers, designers and technical leads dependant on the type, and scope of the project. All the members have related systems development experience, and none have received any additional training for designing Web-based systems. There is a project manager assigned to each project. Other members who are not part of the Web-based design team, but also form part of the development process include Database administrators, Business analysts, and Application and Software testers. On this project the development team consisted of one project manager, and two developers.
6.1.2 The Project

The Web-based system focused on in this study is called: "Namibia Internet Banking System". The project can best be described by the project leader as: "The system was developed to provide banking customers with banking functionality over the Internet". The goal of the project is to provide customers, who have not had the services earlier, with Internet banking. Another goal is to provide a better experience over the Internet, as the previous legacy system had major performance issues as well as to reduce the total cost of ownership for the organization. The project was aimed at users in Namibia who mainly consist of marginal income workers. Initial expectations highlight a user base of around 5000 users, but this expected to increase quickly.

The time available to develop this project was six months, with a strategic and very high importance priority. Various challenges were identified by the development team for the project including:

- Bandwidth limitations, for example a user that might be logging in through a dial-up link, being constrained to this connection.
- Users with old hardware and software which might not include a new version of Windows or Internet Explorer. Standard Bank can not expect a user to download an upgrade which can vary in sizes from five to thirty megabytes.
- A complete new framework had to be designed to overcome the constraints of the legacy system.
- Some size constraints had to be reconsidered as a result of a new version of the application server that was implemented.
- The legacy system was difficult to change with requests taking very long to be completed, and be prone to errors. The costs associated with these changes were also very high, and needed to be reduced.

Security issues which the projects also had to address include a security framework. Within this framework the system had to protect users from various dangers and risks through threat detection, for example protection from a dial-up service attacks.

6.1.3 System Development Methods used

According to the Project leader an in-house developed systems development methodology called the 'BLUE' process was developed by Standard Bank. He
described it as: "We have an overarching process called the BLUE process that identifies inception, definition, construction, test, implication and rollout, so the activities that we did, fell into that. For example, during the inception stage we even pre-incept those user studies that were done, the performance evaluation was done to determine that there is actually a need to change the way the application renders itself on the Internet." This systems development methodology is based on a combination of the well known Waterfall development methodology and the Object Orientated Approach. In short, the team signs off on milestones that have been set forth in the development life cycle.

The system development methodology is summarized next with the sign-off steps described by the project leader, and development team members:

1. **Inception** – During this stage user studies that were conducted are analyzed. Performance evaluations are done to determine whether the system actually needs to change, and the way the application renders itself on the Web.

2. **Definition** – In this phase all the use cases and conceptual work are analyzed, and definitions created. Two of the main contributions in this phase include the Functional Specification Sheet (FSS) and the Technical Specification Sheet (TSS). The FSS is supplied by the organization, and contains mostly the Business rules and Use Cases for the project. The TSS focuses on the Front-end of the system, and includes Class and Sequence diagrams which will show the flow of processes.

As discussed in chapter 4, a methodology can use certain techniques and tools in its development process. The techniques used by Standard Bank include:

- Use Case design
- Class diagrams
- Context diagrams
- Activity diagrams
- Data Flow diagrams
- ERD diagrams
- Sequence diagrams
- Screen Flow diagrams
- UML diagrams
Tools that were utilized during development include Microsoft's Visio®, and IBM's Rational® Software.

3. **Construction** – This phase includes the completion of detailed design. The actual building of all the user interfaces, service components and the backing components. Software is mainly developed on a .NET® platform.

4. **Testing** – Next, the team will gather testing scenarios from the organization to test the system. After this the system is handed over to a separate testing team which draws up test cases using the FSS as a guideline. If the system is accepted, systems integration testing is done. This part of the testing cycle can consist of more than three cycles. The final testing step utilizes user acceptance testing.

5. **Implementation** – The system is installed and run on the applicable internal servers. Full functionality must be available as to ensure that the system is functioning completely. The actual application services are deployed on UNIX® and Solaris, using the IBM® application server.

6. **Roll out** – The final phase in this process is where the implemented system is made live and available to the public to use.

The whole development and implementation of a complete new Internet Banking System took Standard Bank just over five months. There were no reported comebacks or required changes from the initial system. A team member quoted only minor changes required later on: “We had two small fixes that we had deployed subsequently, and as I say, when we extended the system to support South-African Internet banking, we revisited some of those functions just to get the way the system was built, more balanced.”

Standard Bank uses a system development methodology which they do follow and adhere too fully during development.

6.1.4 **Methodology influence**

Standard Bank places high emphasis on testing the applicable systems. Because their methodology allows for various sign-off phases that include different testing cycles, their methodology fits perfectly in the organization’s company policy of ensuring that the Web-based system are easy to access, use and maintain. The methodology also allows Standard Bank to arrange a meeting with the management.
team of the organization at each sign-off phase. By doing so, the management team can easily identify possible problems in the proposed Web-based system before running over budget or time constraints.

The Standard Bank team has quite a good understanding of system development methods that are available, and highlights some of the disadvantages of their approach.

Firstly, because the BLUE process is based on a Waterfall and Object Orientated approach, changes during development are quite difficult to address. A good example is a database that was developed, and in the testing phase, was found to have the wrong naming conventions, or defined fields. With an iterative approach, a change like this might be a lot quicker and easier. Another problem according to the development team is a lack of a proper description of the project requirements. One developer quoted: “There is also a lot of red tape that might not be needed. There is certain things that might be counter productive, the process might not always be clear. There is not always a formal route, and this slows things down”. A suggestion is the creation of a more formal document that lists all the requirements of the project.

Although Standard Bank is satisfied with their current system development methodology, taking into account some of the disadvantages highlighted, it can be noted that Standard Bank should address these issues when revising their current in-house systems development methodology.

6.1.5 Summary

The Standard Bank has developed an in-house systems development methodology based on the traditional Waterfall and Object Orientated approach which they do follow and adhere to during development. The main advantage for the team is that on every phase a sign off is required. This does not only help to motivate the team, but also judge if the project is within the required timeframe or not. A quote from the project team leader describes this aspect best: “I think we succeeded because we had the right level of focus and support within our community to succeed.”

A disadvantage of the current approach is that changes during development are difficult to address and there is a shortage of descriptions on the project requirements.
The BLUE process added to the success of this project, and with each phase addressing all the needs required by their Web-based system, this formal route has added to the growing trust of using such a methodology.

6.2 Findings for Working Webs cc

6.2.1 The Team

The Working Webs team consists only of a team leader and a developer. It is important to remember that the Working Webs organization is a very small company, which mainly develops either Web-based systems, or stand alone Web pages. The project leader describes it as: “Our business takes a less formal approach, we have found that for example if you only need one developer, it is easier to take that approach.” Although Working Webs also provides training to students for Web development, the development team has not received any additional training.

6.2.2 The Project

The Web system discussed in the interview is called the Sykes Race Horse Formgrids. A team member defined it as: “The system is actually a race horse information system”. According to Working Webs, Formgrids was developed to enable the user to gather information as easily possible studying race horses. History and specified horse’s performance can be viewed easily to understand grids, aiding in helping a betting person to track a horse’s odds. Importance set on the project was high, as the client needed the system to be implemented quickly due to competition that could copy the idea.

The goal of the system was to create an easy-to-use interface, and to maintain upload speed as fast possible by limiting bandwidth where necessary. This main goal of the system was to present the information in a useful and different fashion. Another goal was identified by the project leader as: “We must try and keep the interface clean and simple, almost downgrade the user interface, back to the way it was intended, you know, links should be underlined, or at least a different color, so people can work their way around”.

The user base of the system includes public and professional betters, as well as the racehorse owners, who might need a summary of their horse’s performance. The
system which took three months to be developed is still in its infancy, and currently has over 750 registered subscriptions.

The Web-based system was faced with four challenges:

- To produce a front end that is easy to understand and navigation to cater for different user types.
- To face the challenge of interpreting the client's request into a feasible Web-based system.
- To support the client on the approach that needs to be followed for data design to satisfy all the requirements.
- To code formulas to calculate the grid values, and design an interface that will display this grid in an easily understandable way.

Because the Web-based system designed by Working Webs mainly focuses on information sharing, the focus on security was not such an important aspect to integrate into the system. The only security aspect to consider was on creating subscription accounts to allow users to access the information after payment.

6.2.3 System Development Methods used

Working Webs did not follow a specified approach, and development followed the process described next.

Working Webs started its development process with the team having a few meetings with the client. The idea for this initial step was described by a team member as to: "...enable us to understand exactly what the system is about". In this step all the requirements are listed fully. After this another meeting was scheduled with the client, where the team and the client had a brainstorming session about all the requirements, to make sure they were all understood.

In the next step these requirements were used with the team doing lateral thinking, reasoning what user interface would be required for the proposed Web-based system. This step mainly focused on problem solving.

The third step included creating screen flow diagrams. The team called this: "Diagrams showing how the data could be projected on the screens. The different layouts and the movement between these screens". Another diagram included the
data flow diagram showing how the data would interact and be implemented in a database.

In the fourth step, design was done on the proposed system. The developer used the requirements gathered in step one to develop a system that not only satisfied all of these, but also ran all the algorithms from the database to display the correct data in the grids. The project commented that: "If you have a bit of experience you usually get it right the first time, and then you just tweak it a bit in the end to get the speed right." Another meeting was arranged to get feedback from the client about the proposed Web-based system. Changes that were required could be made.

For the final step, the system is implemented, and access granted to external users to test the Web-based system. After this stage, the system is made available to the public to use.

6.2.4 Methodology influence

Unlike the case with Standard Bank, Working Webs do not follow a fixed approach in developing their Web-based systems. The team has knowledge of methodologies, but one team member explains their reasoning as: "Our business takes a less formal approach, I think the more formal approach would suite large corporations better, especially in large groups where you need all the applicable people to talk to each other."

Although this informal process could easily be done by a smaller team, or with less administration, it could also be the apparent reason that their Web-based systems have a lot of changes required after implementation.

6.2.5 Summary

Within the process of developing Web-based systems Working Webs does not follow a specific approach. They mostly gather requirements, and design the system around these requirements. They have knowledge of the system development methods available, but as discussed earlier, believe these methods are not suitable for their organization.
6.3 Findings for Fourier Approach Pty (Ltd)

6.3.1 The Team

The Fourier Approach team consisted mainly of a project manager and two developers. The project leader described the rest of the team as: "There were four main groups. The one group was the business analysts, defining exactly what they want to achieve on a business level, then there was a group system analysts who transform the business requirements into logical requirements. And then there were two development teams, one was developing the Web-based application, and the other team the fat client application". As in the case with the other organization, none of the team members received any additional training for Web-based system development.

6.3.2 The Project

The Web-based system developed by Fourier Approach was called the PROMPT system. The project leader described the Web-based system as a program management system, with programs defined as a set of projects grouped together to manage the strategic goals of a company. The objective of the system was to align the business strategic drives and goals with detailed projects that are being done by the client.

The senior developer described the three parts found in the system: "There is a SQL server database, and then there are two front end suites. The one is a Web-based application, where all the program managers can update all their specific projects, and everything related to the projects, for example, meetings, deadlines, milestones and all that kind of stuff. Then there is also a fat client side where program managers can manage their programs on strategic levels that align to the company goals."

The Web-based system is used by government offices that at any time could run between 3000 and 6000 projects. The Web-based system took between three and four months to be developed. Because the project was completed with an outside partner, high importance was set on the project being completed successfully, and in the designated timeline.

The project had certain requirements that had to be met, which also added to some of the challenges found in the project. These include:
• To develop a Web-based system that must fill the gap between the strategic goals of the company and the projects in the company.
• To develop strategic reports that could be run from the Web-based system.
• To create a user friendly graphical user interface that would fill the needs of both inexperienced and expert users.
• To create a Web-based system that would require minimal upload time for the program that requires a lot of pages to be loaded quickly.

Because the PROMPT system is used by the client only, Fourier Approach only incorporated account and password functionality into the Web-based system as its security function.

6.3.3 System Development Methods used

Fourier Approach designed their own internal methodology to develop their Web-based systems which is based on the popular Waterfall approach. The project leader describes the methodology: "We use a case tool called Casewise®, and in Casewise® we have a methodology that we use to define the specification side of development. And out of that specification we develop the application, Web forms etc". Fourier Approach supplied a summarization of this methodology which can be found in Appendix A.

The methodology has ten phases, with three main sign-off phases that require the client to be included as well. The three sign-off phases include the functional specification phase, logical specification phase, and then the physical specification phase. According to the senior developer: "The physical specification phase includes screens where you can actually see how the screen will look, and how you can interact with that on a screen level and we have the functional specification where there are more high level functions that are going to get into the application." These specifications cover certain stages which the developer explained as: "What we did is we started of basically just to define the requirements on the business level. From that we broke it down into functional requirements, and defining all the functions. From that we broke it down into logical components. That we actually broke down into physical form, and the database design level and all the rules associated with that. From that point onwards we actually used that specification to develop the system."
As discussed with the Standard Bank organization, a methodology can use certain techniques and tools in its development process. Some of the techniques used by Fourier Approach include:

- Screen flow diagrams
- Screen designs
- ERD designs

Fourier Approach also has strict design templates which have to be adhered to. This aids in the organization having a general recognizable "look" to their Web-based systems. The methodology that Fourier Approach utilizes has certain sign-off phases which need to be completed as the steps in the methodology progresses. The client also signs off on these phases, promoting interaction with the developers of the system. This can aid in a successful client acceptance of the system. The methodology utilized by Fourier Approach is adhered to strictly. It is utilized with every Web-based system developed by the organization.

6.3.4 Methodology influence

The main reason Fourier Approach uses their methodology is described best through the words of the project leader: "The main reason why we follow the process is that we saw to many cases in the industry that a development deliverable doesn't really suit the client's requirements. We adopt this methodology to make sure that there is three sign-off phases so that the client exactly knows what is going on in the development process, and don't only see the deliverable at the end of the day. So by following this process the client actually knows what he is going to see at the end of the day. He is involved all the way through the cycle of the methodology, and he actually signs-off on all the stages. Knowing what he is going to get at the end of the day."

Fourier Approach uses a tool called Casewise® to administer their methodology. Developers in the team mentioned that they would rather prefer a straight forward MS Word document than using the Casewise® tool to administer their methodology. The problem is summarized by one of the team members: "The problem is that we do not know how to use it. None of us had training in it, and don't even know how to create a specification in it." But the team members also acknowledge the advantages of such a tool by saying: "Casewise® has its benefits, because when you link projects together you can in theory do stuff that MS Word can't do. For instance if you want to
do an impact analysis, you change a field, and it will affect the whole scope for you automatically."

Fourier Approach uses this methodology in the development of all their Web-based systems. The only cases where it is not used include additions or changes that need to be made to already implemented systems.

6.3.5 Summary

Fourier Approach has designed an in-house methodology to use when developing Web-based systems. Fourier Approach places high importance on a strict adherence to their methodology.

The development team praises their success rate of developed Web-based systems to their methodology that interacts with the client through the whole development process. This minimizes end rejection by the client, who knows exactly what to expect in terms of the completed system.

6.4 Results of cross-case analysis

The goals of this research were as follows: Firstly, to describe the context in which Web-based systems are being developed. Secondly, to describe how system development methodologies are used in practice to develop Web-based systems and to what extent it is used, and to describe the influence of system development methodologies on the quality of Web-based systems.

In the following section the results from the cross-case analysis method will be discussed. The table created from the case study data which were used for cross-case analysis can be found in Appendix B.

Propositions can be seen as patterns that match in the data between the different case studies. As highlighted by Palmquist (1997), these patterns act as evidence to support the findings. When the data does not match, a deeper investigation can be addressed to identify the reason for the conflict. For easier identification on the propositions and contradictions discussed next, the organizations will be referenced where applicable as: SB (Standard Bank), WW (Working Webs) and FA (Fourier Approach).
6.4.1 Context

In this section the context in which these Web-based systems are developed will be discussed.

It is interesting to note that none of the organizations sent any of their development team members for training on how to develop Web-based systems (SB, FA and WW). All the developers only had previous experience in developing traditional systems.

Both Standard Bank and Working Webs have placed high importance on their Web-based system’s goals which aim to limit bandwidth usage, and increase the speed of their Web-based systems. The contradiction found in terms of Fourier Approach is that they do not place attention on these issues at all. The reason for this is that both Standard Bank and Working Webs have a Web-based system that is implemented on the Web, and they have to address these bandwidth limitations. Fourier Approach developed a Web-based system that is mainly accessed on an internal network by the client, with external access being used only some of the time. This places minimal bandwidth constraints on Fourier Approach’s Web-based system.

The user base of the identified Web-based systems differ from the smallest with a 750 subscription base (WW), to a Web-based system that expects over 5000 clients in the near future (SB). Fourier Approach designed a Web-based system that needs to manage anything between 3000 and 6000 projects at any given time.

Typical time for development for a fully implemented Web-based system can be expected to last between three to four months (SB, WW and FA). All the organizations have noted that their projects had to adhere to these set timelines for the whole Web-based system development process. During the development process, the two larger organizations put a big emphasis on user involvement (SB and FA). Both organizations explained that the reason for this is to limit the changes that might be required after implementation. In contradiction, the one organization Working Webs which has not involved the user more in the development process has stated that they have a lot of changes that are required after the implementation of their Web-based systems.
The importance set on the projects was all rated as "high" for all three organizations (FA, SB and WW). The results for Fourier Approach and Working Webs could be explained by the fact that competitive pressures in which these organizations operate may mean that any advantage is short-lived and will be copied quickly.

Security requirements are another aspect which required attention. Standard Bank was the only organization who placed huge emphasis on this necessity. Fourier Approach and Working Webs also made use of security measures, but these were standard generalized approaches. Because Standard Bank is an Internet Banking service provider, data used by the Web-based system is classified as extremely sensitive, and explains easily why Standard Bank needs to not only do extensive testing, but also make sure the security requirements are fully satisfied.

All the organizations have highlighted that the biggest difference for them between a traditional system and a Web-based system is the rich interface and attention the latter requires. They have also placed high importance on the bandwidth limitations that are applicable to Web-based systems. This aspect has been identified as critical to the sustainability and success of the Web-based systems (SB, WW and FA).

6.4.2 System development methodology use

The two larger organizations follow their own in-house development methodology with both based on the Waterfall approach (SB and FA). Standard Bank also utilizes the Object Orientated approach in its methodology. The smaller organization, Working Webs, does not follow any formal approach when developing Web-based systems, but acknowledges that the more formal approach would be better suited for larger companies in general. None of the organizations use a methodology that focused mainly on Web-based systems development.

The two larger organizations have identified the main advantage of their methodology as the sign off phases that can be included on each step of the methodology's process (SB and FA). This enables the identification of milestones that have to be reached by certain deadlines, as well as to enable the inclusion of outside parties when needed. In the case of Fourier Approach this will be the client of the Web-based system, and with Standard Bank, the management team that needs to report on progress to the rest of the organization. The results indicate that the system's development methodology supports project management in these organizations.
Both organizations adhere strictly to their in-house methodologies (SB and FA). Both organizations only deviate from their methodology when small changes to the implemented Web-based system are required. This was the only adaptation identified that was applicable to the methodologies. It is interesting to note that with both these organizations management administers a strict following on their defined methodologies.

The smaller organization, Working Webs, has displayed knowledge of methodology usage, and admitted that it would just not be applicable to their organization. Both Standard Bank and Fourier Approach also have a good understanding of the different methodologies available, and both highlighted the advantages the methodologies give when developing larger Web-based systems. Both also are content with following the methodological approach, and won’t replace the approach any time soon.

6.4.3 Influence

Albertyn (2002) remarked that the design and development of Web-based systems have increased in both complexity and challenges faced as a result of the changing environment of the Web. Albertyn (2002) also noted that the biggest challenges are compressed time schedules, build-or-buy decisions and having to incorporate to traditional legacy systems.

The two larger organizations both claim that by using methodologies in their development processes the quality of the developed Web-based systems have increased (SB and FA). Both organizations adopted a methodology that suits their organization’s project management needs completely, and with this management functionality, can easily face the challenges defined by Albertyn (2002) in the beginning of this section. Each organization’s methodology also fits in perfectly with the requirements expected by their individual development processes.

Fourier Approach uses an in-house methodology that delivers various sign-off phases during development. This incorporates the users of the system and aids in identifying problems and issues before the next step in the development process starts. Because Fourier Approach delivers Web-based systems in the commercial market, another advantage of the sign-off phases is that these protect them from
requests by the client for changes in the development cycle. Using the same sign-off phases in their development process as well, Standard Bank can arrange cyclic meetings with the management team of the organization for each sign-off phase. By doing so, the Standard Bank management team can easily identify possible problems in the proposed Web-based and approach these before the project runs over budget or time constraints.

A disadvantage faced by Fourier Approach is the Casewise tool they utilize to administrate their methodology. Team members did not receive any training on this tool, and battle to utilize the tool fully. The disadvantage of the methodology followed by Standard Bank is that because each phase is signed-off, changes required later on are difficult to make and to administer.

In contrast with the larger organizations, Working Webs did not following any formal process at all. A working prototype is created after various requirement sessions with the client. After this step periodical meetings are arranged with the client, where changes and improvements on the Web-based system are discussed. These will then be addressed until the next meeting, until the client is satisfied with the final Web-based system. The main disadvantage of this informal approach is that Working Webs have a lot of changes that are required after implementation of the Web-based system.

Although it can be argued that because Working Webs is a smaller organization the formal approach might not be suitable, the advantages realized by Fourier Approach and Standard Bank contradict this statement. By using a developed in-house methodology which fits in perfectly with the organization's goals, and business processes, a suitable methodology could just as easily be adopted by Working Webs. The advantage of this for Working Webs could be the increased communication imposed by these methodologies. This helps to identify any issues and problems which could arise easily.
6.5 Summary

In this chapter, findings derived from the data analysis phase were discussed for each organization individually. For each organization this included a discussion on the project's background, the development team, the development process followed, and the awareness of the organization in terms of developing Web-based systems with the use of a formal development process.

In the next section the research findings were identified. Results of the cross-case analysis were presented, with focus being placed on the context of the individual projects, the system development methods used, and lastly the influence of the methodologies on the development process were discussed.
Chapter 7

Findings and Contributions

The general goal of this study was to investigate how Web-based applications are developed in South Africa. Although formalized Web application development methods have been proposed by the academic community, Barry and Lang (2001) concluded from their research on Web and multimedia systems development that 24.6% of general companies do not use any methodologies at all.

With the number of businesses that utilize the Web for their daily activities growing daily and Web-based systems increase in popularity, there is clearly a need for further research in this area. Studies that focus on how Web-based systems development methodologies are used to develop Web-based systems in South Africa are very limited. Furthermore there are no indications of the approaches organizations and businesses use to develop these Web-based systems in South Africa.

This chapter will summarize the contributions of the research done in this study, and elaborate on its limitations and possible future research into the use and effectiveness of Web-based system development methodologies.

7.1 Research contributions

The general goal of the study was to investigate how Web-based systems are being developed in South Africa. This was achieved by addressing the following goals:

1. Analyze and summarize a selection of different Web-based system development methodologies available for development.

Five Web-based systems development methodologies were discussed. The methodologies included: Web IS Development Methodology (WISDM), Internet Commerce Development Methodology (ICDM), Web Engineering, Extreme Programming and the Relationship Management Methodology (RMM).
2. Compare these system development methodologies according to a framework of Avison and Fitzgerald (2003).

After discussing the five methodologies, a comparison was done to highlight the main differences and similarities between these methodologies. The methodologies were compared according to the framework suggested by Avison and Fitzgerald (2003). An important finding was that even though all the methodologies have the same end goal in sight, nearly all of them are based on a different philosophical approach. The philosophical approach followed by each methodology can be summarized as:

- **WISDM** follows a soft system approach that covers both analysis and design of the system. The MultiView framework provides a basis for constructing a situation specific methodology.
- **RMM** is a data driven methodology most suited to applications that have a regular structure. The methodology aims at being a basis for the design and development of robust hypermedia applications.
- **Extreme Programming** is a flexible approach of software development based on underlying principles of simplicity, communication, feedback and courage.
- **Web Engineering** follows a scientific approach that advocates a process and systematic approach to development of Web-based systems.
- **ICDM** has a paradigm, aimed at business-to-consumer application development that is intensely social in nature. ICDM also provides a guiding framework for developers.

3. Evaluate theoretically the selection of system development methodologies for their suitability to develop Web-based systems.

The theoretical evaluation indicated that the two characteristics that all the methodologies adhere to, both have to do with information integration. These two characteristics were the integration of various information bases and the support of networks for information access. A lot of emphasis is also placed by organizations on the adherence to time pressure by these development processes. This is one characteristic which less than half the methodologies took into consideration.

Based on the information derived from this theoretical evaluation of the selection of system development methodologies, it seems that the Web Engineering
methodology satisfies most of the characteristics associated with Web-based systems.

4. Describe the context in which Web-based systems are being developed.

After analyzing the data from all the organizations, it was clear that none of the development team members have been for any additional training in developing Web-based systems, but the developers had previous experience in traditional information systems development.

The study also found that when a Web-based system is run purely on a Web platform, bandwidth usage is highlighted as the biggest challenge for the project. If a Web-based system is more focused on an intranet platform, with minimal access needed through the Web, bandwidth constraints are not even considered a priority.

Another finding of the study was that a typical user base of a Web-based system can range from a small commercial Web-based system with 750 users, to a self service system with over 450 000 registered users.

The development time for a fully completed Web-based system can be expected to last anything between three to four months. This development time was the same for all the organizations studied, and was not influenced by the size of the Web-based system.

All the organizations placed a high priority on the importance of the Web-based system. One of the reasons was that the organizations needed new functionality from existing Web-based systems. Another reason was that competitive pressures caused the Web-based systems to be completed quickly. This finding coincided with Albertyn (2002) that Web-based system faced challenges daily as a result of the changing environment of the Web.

All the organizations have placed emphasis on the fact that the biggest difference between traditional information systems development and Web-based systems development is the extra attention required for the rich interface requirement of Web-based systems.
5. Describe how system development methodologies are used in South-Africa to develop Web-based systems, and to what extent it is used.

The development methodologies used for Web-based system development in South Africa are not methodologies specifically created for Web-based development. Methodologies are rather developed in-house to comply with all the needs of the organization in terms of Web-based systems development.

The larger organizations in South Africa do follow methodologies to develop Web-based systems. These organizations adhere strictly to the methodology, and its development process. The smaller organizations have knowledge about these formal approaches, but choose to use no systems development methodology at all. This finding can be supported by Barry and Lang (2001) who defined a similar situation that although some Web-based systems development methodologies exist that can be used to address these problems, e.g. WISDM, ICDM, e-RAD, Web-engineering, and WebXstream, practitioners are not using them.

6. Describe the influence of system development methodologies on the quality of Web-based systems.

All the organizations that use system development methodologies to develop Web-based systems have highlighted that the main advantage of using a methodology is the project management aspect associated with it. This does not only help to increase the quality of the Web-based systems developed, but also help the project to stay within the planned scope and budget.

Another advantage is the improved communication that the methodology brings not only in the development team, but to outside partners as well.

7.2 Limitations of the study

Although the three projects discussed in this study were representative of different categories of Web-based systems, it should be noted that these projects might not represent the population as a whole, and it would not be advisable to generalize the results of this study.

One of the main reasons for this limitation is that most of the large organizations are skeptical in supplying information to an outside source. Even though the research
goal was explained, and anonymity was promised, few organizations were still keen on allowing this researcher to have interviews with their staff.

Another limitation for this study was the amount of information which could be easily gathered on Web-based system development methodologies. This limitation could be overcome by acquiring access to more study materials, or to source more information from the developer of the methodology.

7.3 Future work

This study could be followed up with grounded theory and quantitative research to substantiate the results even further.

Another research topic to consider would be to investigate why organizations do not use the Web-based system development methodologies discussed in chapter 3. Are there reasons for this, or do these organizations just not know about these methodologies, or might it be that these organizations do not believe these Web-based systems development methodologies would work?

Current trends in Web-based system development suggest that the development of these systems might be more dependant on the type of software platform used to the physically design the system than the method followed itself. With the increase of programs that have built in Web design templates, and intelligent coding backbones, the attention could shift more on the software supporting the development process. It would be advantageous to do research on this trend, and to conclude if that is really the case.

Another research option could be to investigate the differences between traditional information systems and Web-based systems in detail, as to defer if a process could be identified to convert the traditional information system to the latter. Because a lot of organizations are considering the Web-based system option, it could be of value to investigate if such a transition could be possible, and feasible to approach.
References


Kalanjee, V. 2006. The use and effectiveness of systems development methodologies in South Africa. Potchefstroom: PU for CHE.


## Appendix A

<table>
<thead>
<tr>
<th>Phase Names</th>
<th>Description</th>
<th>Processes (Steps)</th>
<th>Outcomes</th>
<th>Templates</th>
</tr>
</thead>
</table>
| 1 Preliminary Investigation Phase (Initial study phase, survey phase, or planning phase) | The preliminary investigation defines the scope of the project and the perceived problems, opportunities and directives that triggered the project | List Problems, Opportunities and Directives  
Negotiate Preliminary Scope  
Assess Project Worth  
Plan the Project  
Present the Project and Plan | Project Charter  
Presentation of Project Charter to Steering Committee  
Project Plan | Project Charter  
Problem Statement  
Diagram  
Project Plan |
| 2 Problem Analysis Phase | The Problem Analysis Phase takes an in-depth look at the current system or manner in how the work is done. | Study the Problem Domain  
Analyze Problems and Opportunities  
Analyze Business Processes  
Establish System Improvement Objectives  
Update the Project Plan  
Present Findings and Recommendations | Updated Problem Statement  
Cause-and-Effect Analysis Results  
Correct Process Models and Process Analysis  
Updated project plan and scope | Cause-and-Effect Analysis Diagram  
Project Plan |
| 3 Requirement Analysis Phase | The purpose of requirements analysis is to identify the data, process, and interface requirements for the users of the new system | Define Requirements  
Draft functional and non-functional | Updated & Signed URS |
<table>
<thead>
<tr>
<th>Phases</th>
<th>Description</th>
<th>Analyze Functional Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Business Requirement Statement (URS)</td>
</tr>
<tr>
<td>4</td>
<td>Decision Analysis Phase</td>
<td>Data and Process Models</td>
</tr>
<tr>
<td></td>
<td>The purpose of this phase is to identify candidate solutions, analyze those candidate solutions for feasibility, and recommend a candidate system as the target solution to be designed</td>
<td>Project Plan &amp; Budget</td>
</tr>
<tr>
<td>5a1</td>
<td>Functional Design Phase - Build Solution</td>
<td>Design the Application Architecture (Context Diagram, Data Flow Diagrams)</td>
</tr>
<tr>
<td></td>
<td>The Functional Design of the System</td>
<td>Design the System Interface (Design Preliminary System Screen Layout)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Design the Logical Database Schema</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Functional Decomposition Diagram</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Functional Spec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Functional Test Plan &amp; Results</td>
</tr>
</tbody>
</table>
| 5a2 | Logical Design Phase - Build Solution | The Logical Design of the System, databases and applications | Process Modeling  
Construct a Functional Test Plan  
Handover Work Session | Logical Design Specification  
Logical Test Plan  
Updated Project Plan and Budget | Logical Design Specification  
Logical Test Plan  
Updated Project Plan  
Updated Budget  
LQS Budget |
|---|---|---|---|---|---|
| 5b | Design Phase - Buy Solution | The purpose is to identify and research specific products that could support our recommended solution, solicit, evaluate, and rank vendor proposals, select and recommend the best vendor proposal and contracting the awarded vendor. | Research Technical Criteria and Options  
Solicit Proposals (or Quotes) from Vendors  
Validate Vendor Claims and Performances  
Evaluate and Rank Vendor Proposals  
Award (or LET) Contract and Debrief Vendors | A contract with the selected vendor which had the winning proposal | Vendor Proposal Ranking |
<table>
<thead>
<tr>
<th></th>
<th>Construction Phase</th>
<th>System Construction is the development, installation, and testing of system components</th>
<th>Build and Test Network, Setup Development Environment (if Necessary)</th>
<th>Build and Test Physical Database</th>
<th>Install Software Packages (if Necessary)</th>
<th>Write New Programs</th>
<th>An unpopulated database</th>
<th>New programs</th>
<th>Functional Test Plan</th>
<th>Functional Test Plan (Signed)</th>
<th>Logical Test Plan</th>
<th>Logical Test Plan (Signed)</th>
<th>Unit Test Plan</th>
<th>Detailed Conversion Plan</th>
<th>Training Manual</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Testing Phase</td>
<td>System Testing is the functional and logical testing of the developed system</td>
<td>Functional Testing of New System</td>
<td>Logical Testing of New System</td>
<td>Tested System</td>
<td>Functional Test Plan (Signed)</td>
<td>Logical Test Plan (Signed)</td>
<td>Unit Test Plan</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Implementation Phase</td>
<td>System Implementation is the delivery of the developed system into production</td>
<td>Conduct System Test</td>
<td>Prepare Conversion Plan</td>
<td>Install Databases</td>
<td>Train Users</td>
<td>Convert to New System (Deployment Process)</td>
<td>A new database populated with the old system data</td>
<td>User training and documentation</td>
<td>An operational system implemented in the business</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Phase</td>
<td>Description</td>
<td>Additional Steps</td>
<td>Product</td>
<td>Documentation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-------------------------------------------------------</td>
<td>----------------------------------------------</td>
<td>---------------------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9   System Maintenance Phase</td>
<td>System Maintenance is performed to identify and correct the errors and bugs that slipped through the testing phase</td>
<td>Validate the Problem</td>
<td>A tested and corrected system into the live environment</td>
<td>Acceptance Sign-off Document</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10  System Enhancement Phase</td>
<td>System enhancements require the System Analyst to evaluate the new requirements or to direct the enhancement processes through the required phases</td>
<td>Analyze Enhancement Request</td>
<td>A Candidate program that is reengineered and ready for production</td>
<td>Enhancement Template</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Appendix B

<table>
<thead>
<tr>
<th>INTERVIEWEE</th>
<th>Working Webs</th>
<th>Fourier</th>
<th>Standard Bank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interviewee Background</td>
<td>I see my activity which is to develop software.</td>
<td>0</td>
<td>Contractor, I just came and started working here...</td>
</tr>
<tr>
<td>Interviewee experience</td>
<td>0</td>
<td>0</td>
<td>3 years.</td>
</tr>
<tr>
<td>Interviewee Role</td>
<td>0</td>
<td>I take a senior developer role.</td>
<td>Senior designer/developer/technical lead. I'm a designer/Developer.</td>
</tr>
<tr>
<td>Training/Not</td>
<td>0</td>
<td>We didn't do any direct training. Case wise, The problem is that we do not know how to use it. None of us had training in it...</td>
<td>They don't usually send us on training. No additional training.</td>
</tr>
<tr>
<td>PROJECT</td>
<td>Guys that does the betting on the street. Racehorse owners as well call the information and use it. 750 odd subscriptions.</td>
<td>Around 30-50 users per site. One is the program manager. The project manager that updates the projects for us...</td>
<td>Farmers and working class people. Marginal income customers. Probably in the 5000 range. I think it's about 440 000 now.</td>
</tr>
<tr>
<td>User Base</td>
<td>In this case, I think not.</td>
<td>The marketing company that was defining the requirement, if they are seen as a user, they were part of the specification and testing phase. Client actually knows what he is going to see at the end of the day.</td>
<td>In the sense, our customers, as a development team was the business community. Plus the people in our contact centre. Customers from our business unit. Whole bunch of meetings with business people. Business is supposed to give a clear view of what they want the system to do. They usually involve the technical people. Sitting with the business people namely the manager.</td>
</tr>
</tbody>
</table>
Project Adherence

Have done a few subscription systems at the moment. And for him, the system was nothing unless it is perfect; as far as he's concerned the way we displayed was light-years ahead of any thing close to it.

Project Architecture

Have done a few subscription systems at the moment. And for him, the system was nothing unless it is perfect; as far as he's concerned the way we displayed was light-years ahead of any thing close to it.

Project Challenges

Basic requirements were a lot of formulas running on a spreadsheet. Limited bandwidth. PC that does not necessarily have the latest version of Windows or Internet Explorer. You can't dictate what the customer must have or not have. Challenge we had is that we were building on a new framework, we also had to overcome in the same space.

Project Description

The idea actually came from the client, it was actually just designed to, on one screen you can get such a good view of - what, for a particular race, let's say the race is coming now, what were the previous performances, and then you can track that very quickly, just by clicking on the race.

There is no real good configuration tool to keep configuration over source code. Lot of small pages that are that is linked together. The system was developed to provide banking customers with banking functionality over the internet. But it was limited to movement of money between accounts. And things like that, not for application of loans or long running processes. We did provide 60% of the functionality.
| **Project Goal** | The goals of the system are to align the business strategic drives, with umm... The detailed projects that are happening. The system is designed to align the strategic goals to the detailed projects. We develop a system that fills the gap between the strategic goals of a company, and detailed programs, and projects in a company. |
| **Project Maintenance** | We have a help desk where the user and call in and log a problem. I would say you would need about 10 to 20% time of one guy on the team... and ours is just a small group of that... |
| **Project Manager** | 0 |
| **Project Name** | Sykes Race Horse form grid, | Namibia Internet Banking system. |
| **Project Outcome** | Strategic reports that were developed out of the system, and the user friendliness of the system. It was from a technical viewpoint, but not from a commercial viewpoint. I think it was a success... It has been working the whole time. |
| **Project Platform** | The system is judged on this interface, even more that the functionality within the system. |
| **Project Priority** | The only failure would be if we moved too slowly. The importance was quite high, and there was very strict timeliness. |
| **Project Scope** | About 3 months... 3-4 months... |

That specific development goal was to provide customers with internet banking, who hadn’t had it previously. And the other goal was to provide a better experience over the internet, because the previous version of the system had some performance issues. Improved experience, total cost of ownership would come down.
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Technical Constraints</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Legacy System</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PROJECT TEAM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Database</td>
<td>We had to redesign a lot of the database... database back end... Database design level and all the rules associated with that...</td>
<td>Database descriptions, Data Models, Database structure now could be done better.</td>
</tr>
<tr>
<td>Developers</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Technical Lead</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Testers - App</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Role Diversity</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Analysts</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SDM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>In-house SDM</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

128
<table>
<thead>
<tr>
<th>In-house SDM Adherence</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Design/Construction</strong></td>
<td>Yes, we follow it; I would say about 90%. The main reason why we follow the process is that we saw to many cases in the industry that a development deliverable doesn't really suit the clients requirements. We follow a strict methodology, and involve the client all the way.</td>
</tr>
<tr>
<td>In-house SDM Definition</td>
<td>0</td>
</tr>
<tr>
<td><strong>Implements/Sign Off</strong></td>
<td>...called definition were basically all the used case work and the conceptual work.</td>
</tr>
<tr>
<td>In-house SDM Description</td>
<td>0</td>
</tr>
<tr>
<td><strong>Inception</strong></td>
<td>We use a case tool called Casewise, and in Casewise we have a methodology that we use to define the specification side of that. The one is basically the functional specification phase, then the logical specification phase, and then the physical specification phase.</td>
</tr>
<tr>
<td><strong>Design/Construction</strong></td>
<td>And then I started with the basic user layout. <strong>Design...</strong></td>
</tr>
<tr>
<td><strong>Function, then logical and then finally your physical design.</strong></td>
<td>Construction would be the completion of your detailed design.</td>
</tr>
<tr>
<td><strong>Inception</strong></td>
<td>Usually I start of by listing all the requirements...</td>
</tr>
<tr>
<td><strong>Testing signed off you then release.</strong></td>
<td>During the inception stage we even pre-incept those user studies that were done.</td>
</tr>
<tr>
<td>In-house SDM Reconsideration</td>
<td>0</td>
</tr>
<tr>
<td>In-house SDM Test</td>
<td>Testing by other external users</td>
</tr>
<tr>
<td>Functional Decomposition</td>
<td>0</td>
</tr>
<tr>
<td>Non-Functional Requirements</td>
<td>0</td>
</tr>
<tr>
<td>Requirements Verification</td>
<td>And then what we do is to get the requirements listed fully.</td>
</tr>
<tr>
<td>Screen Interface</td>
<td>My view is that the interface is often too complicated for a lot of people, so at the end of the day they can't track what is actually going. A lot of people I had talked to talked about the site being too busy.</td>
</tr>
<tr>
<td>SDM Awareness</td>
<td>0</td>
</tr>
<tr>
<td>Struts framework</td>
<td>0</td>
</tr>
<tr>
<td>Technical Specification Sheet</td>
<td>0</td>
</tr>
<tr>
<td>Test Environment</td>
<td>0</td>
</tr>
<tr>
<td>Testing</td>
<td>0</td>
</tr>
<tr>
<td>User Requirements Specification</td>
<td>0</td>
</tr>
<tr>
<td>Front End</td>
<td>0</td>
</tr>
<tr>
<td>Back End</td>
<td>0</td>
</tr>
<tr>
<td>TECHNIQUES AND TOOLS</td>
<td>Activity Diagrams</td>
</tr>
<tr>
<td>Class Diagrams</td>
<td>0</td>
</tr>
<tr>
<td>Code</td>
<td>But, it is programming like everything else... 0</td>
</tr>
<tr>
<td>Context Diagram</td>
<td>0</td>
</tr>
<tr>
<td>Data Flow Diagram</td>
<td>Data flow diagram. 0</td>
</tr>
<tr>
<td>ERD Diagrams</td>
<td>0</td>
</tr>
<tr>
<td>Screen Flow Diagram</td>
<td>Screen flow diagram. 0</td>
</tr>
<tr>
<td>Use Cases</td>
<td>0</td>
</tr>
<tr>
<td>UML Diagrams</td>
<td>0</td>
</tr>
<tr>
<td>Sequence Diagrams</td>
<td>0</td>
</tr>
<tr>
<td>-------------------</td>
<td>---</td>
</tr>
</tbody>
</table>

**END RESULT**

<table>
<thead>
<tr>
<th>Ideal SDM</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
</table>

I think the more formal approach would suite large corporations better.  

<table>
<thead>
<tr>
<th>Implemented SDM</th>
<th>I think the more formal approach would suite large corporations better.</th>
<th>0</th>
</tr>
</thead>
</table>

So we did not really deviate. Maybe it would help if they could implement some formal type of document that will list everything that the project requires. And allow for iterations. So that there isn't just a lot of signed fix offs. So for instance if changes need to be made to the FSS document it would be possible. So maybe they should have a proper structure like meetings and talk about progress etc. Maybe also include the business people more in development. Have like periodic refreshment on what the project needs.

<table>
<thead>
<tr>
<th>System's success</th>
<th>Most of the projects that have been failure were not because the idea was bad, but because it was not carried through to the end. That was usually the main reason for failure.</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Strategic reports that were developed out of the system, and the user friendliness of the system. It was from a technical viewpoint, but not from a commercial viewpoint. I think it was a success... It has been working the whole time.</td>
<td>---</td>
</tr>
</tbody>
</table>

Yes, most of them have been fairly acceptable.

<table>
<thead>
<tr>
<th>Iterations</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
</table>

User interface that we took through iterations. Iterations to that document.
Adaptation

Adaptation for Web-Apps

So what we also try and do is actually, to almost downgrade the user interface, back to the way it was intended. You know, links should be underlined, or at least a different color, so people can work their way around. They very rarely can be spec properly from the very beginning. It's a different way of doing things, there are some quite key differences.

In our system there were for example a lot of changes required...

But this was because the client was unaware of the possibilities of the system, and new requirements started emerging...

Where we don't follow it is where the requirement is to add or change a field or make small changes.

Because on the Web you only have certain tools to use, aimed at a non rich interface.

No, not really.

And changes were made for a few months after that.

None, only a few callbacks for enhancements and stuff.

There was a lot of changes through the testing cycle, but after sign-off, none.

Most of the stuff were new requirements, reports modules etc.

Most of the time what happens is that the client can't picture a system until it is in production for at least a month.

We revisited some of those functions just to get the way the system was built more even. Because there is a lot of data that needs to be worked on which we didn't bargain for...

Changes after Implementation

Changes after Implementation - Reason