

**THE USE OF SYSTEMS DEVELOPMENT  
METHODOLOGIES IN MOBILE  
TELECOMMUNICATION SOFTWARE  
DEVELOPMENT IN SOUTH AFRICA**

**C.H. OTTO  
HONS. B.SC.**

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

Supervisor: Prof. H.M. Huisman

May 2007

## **Acknowledgements**

To my grandmother Ita Hamman. I am who I am today because of your love. Thank you for always being there, for keeping me on the right path and helping me to see and experience all that is good in this world.

Thank you to loved ones and friends without whose help, guidance and prayers, I would be lost.

Special thank you to Professor Magda Huisman, your patience, leadership and wisdom is an inspiration to me.

Most of all I thank God for the strength He has given me. Through God, nothing is impossible.

## ABBREVIATIONS

### A

ASN.1	Abstract Syntax Notation
ATP	Acceptance Testing Procedure

### C

CASE	Computer Aided Software Engineering
CORBA	Common Object Request Broker Architecture

### D

DFD	Data Flow Diagram
DSDM	Dynamic Systems Development Method

### E

ETHICS	Effective Technical and Human Implementation of Computer-Base Systems
--------	---

### G

GPRS	General Packet Radio Service
GSM	Global System for Mobile Telecommunication

### H

HTML	Hyper Text Markup Language
HU	Hermeneutic Unit

### I

ICASA	Independent Communications Authority Of South Africa
IE	Information Engineering
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical And Electronic Engineers
ISO	International Standards Organisation
ITU	International Telecommunication Union

### J

JMRAD	James Martin's Rapid Application Development
-------	--

### M

MDA	Model Driven Architecture
MMS	Multi Messaging System
MS	Microsoft
MSC	Message Sequence Charts

**O**

OMT	Object-Oriented Modeling Technique
OOA	Object-Oriented Analysis

**P**

PD	Primary Document
PMBOK	Project Management Body Of Knowledge

**R**

ROI	Return On Investment
RUP	Rational Unified Process

**S**

SAM	Synthesized Architecture Model
SCM	Software Configuration Management
SDL	System Description Language
SDM	Systems Development Methodology
SDP	Software Development Process
SMS	Short Message System
SSADM	Structured Systems Analysis and Design Method
SSM	Soft Systems Methodology
STRADIS	Structured Analysis, Design and Implementation of Information Systems

**T**

TTCN	Testing and Test Control Notation
------	-----------------------------------

**U**

UAT	User Acceptance Testing
UML	Unified Modeling Language
USSD	Unstructured Supplementary Service Data

**V**

VAS	Value Added Services
VB	Visual Basic

**W**

WAP	Web Access Protocol
-----	---------------------

**X**

XP	Extreme Programming
----	---------------------

**Y**

YSM	Yourdon Systems Method
-----	------------------------



## **Abstract**

This study aims to identify the ways in which mobile telecommunications software systems are developed in practice in South Africa. The main objectives are to identify the systems development methodologies used, if any, during development, to highlight certain key aspects of mobile telecommunication software development and identify a list of necessities a systems development methodology should contain when used to develop mobile telecommunications software.

In the highly competitive market of mobile telecommunications, it is extremely important that mobile telecommunications systems are developed in minimal time, with minimal cost while still ensuring quality and usability. As stated by Fitzgerald (1996), a big factor that could have a positive impact on the development process is the use of a systems development methodology. Because of the importance of systems development methodologies, this study identifies and discusses different types of systems development methodologies from the available literature and also three systems development methodologies designed specifically for developing mobile telecommunication systems.

To determine how mobile telecommunication systems are developed in practice, a qualitative research method was used. Case studies were done at three different mobile telecommunications software development companies in South Africa. Data were collected by conducting semi structured interviews with developers or project managers at each of the companies. Coding of the data was done with the help of ATLAS.ti and analysed by means of cross-case content analysis.

It was found that mobile telecommunications software development companies in South Africa tended to make use of in-house developed systems development methodologies, focusing on flexibility and speed of development. This was driven by the constant interaction of software with other telecommunications systems and the speed at which mobile telecommunication software need to be developed. Key aspects of these in-house systems development methodologies were identified as well as a list of necessities which should be addressed during mobile telecommunication system development.

## OPSOMMING

Hierdie studie beoog om die wyse waarop mobiele telekommunikasie sagtewarestelsels in die Suid-Afrikaanse praktyk ontwikkel word te identifiseer. Die hoof doelwitte is om die stelselontwikkelingsmetodologieë wat gebruik word, indien wel gebruik, gedurende stelselontwikkeling te identifiseer. Dit het ook ten doel om klem te lê op sekere elemente van mobiele telekommunikasiestelsel ontwikkeling en om 'n lys van noodsaaklikhede wat 'n stelselontwikkelingsmetodologie moet bevat wanneer dit gebruik word vir mobiele telekommunikasiestelsel ontwikkeling, op te stel.

Omdat die mobiele telekommunikasie-industrie deel van 'n baie kompetender mark is, is dit belangrik dat sagtewarestelsels ontwikkel word in die minimum teen minimale koste, terwyl die kwaliteit en bruikbaarheid van die produk nie afneem nie. Die belangrikheid van stelselontwikkelingsmetodologie word beklemtoon deur Fitzgerald (1996), in dat die gebruik daarvan 'n positiewe invloed kan hê op die stelsel ontwikkelingsproses. Weens die belangrikheid van stelselontwikkelingsmetodologieë, word daar verskillende tipes stelselontwikkelingsmetodologieë in die studie geïdentifiseer en vanuit die literatuur bespreek. Daar word ook 'n bespreking gegee van drie stelselontwikkelingsmetodologieë wat spesifiek ontwikkel is vir gebruik in die mobiele telekommunikasie-industrie.

Om te bepaal hoe mobiele telekommunikasiestelsels in die praktyk ontwikkel word, word daar gebruik gemaak van kwalitatiewe navorsingsmetodes. Gevallestudies is gedoen by drie verskillende mobiele telekommunikasiestelsel ontwikkelingsmaatskappye in Suid-Afrika. Inligting is by die maatskappye versamel deur semi-gestruktureerde onderhoude te voer met ontwikkelaars of projektebestuurders by die betrokke maatskappye. Die inligting is daarna gekodeer met behulp van die rekenaarprogram ATLAS.ti en analise is gedoen deur middel van die "cross case" inhoudsanaliese metode.

Daar is gevind dat mobiele telekommunikasiestelsel ontwikkelingsmaatskappye in Suid-Afrika meestal gebruik maak van selfontwikkelde stelselontwikkelingsmetodologieë, wat veral fokus op aanpasbaarheid en ontwikkelingspoed. Dié word genoodsaak deur die interaksie tussen sagteware wat ontwikkel word met ander telekommunikasiestelsels, asook die spoed waarteen mobiele telekommunikasiestelsels ontwikkel moet word. Die belangrike elemente van die

selfontwikkelde stelselontwikkelingsmetodologieë word geïdentifiseer, sowel as 'n lys noodsaaklikhede wat 'n stelselontwikkelingsmetodologie moet bevat wanneer dit gebruik word in mobiele telekommunikasiestelsel ontwikkeling.

# CONTENTS

## CHAPTER 1

<b>1.</b>	<b>INTRODUCTION.....</b>	<b>1</b>
<b>1.1</b>	<b>Introduction.....</b>	<b>1</b>
<b>1.2</b>	<b>Research and objectives .....</b>	<b>1</b>
<b>1.3</b>	<b>Telecommunications Systems .....</b>	<b>3</b>
<b>1.4</b>	<b>Qualitative Research.....</b>	<b>4</b>
<b>1.5</b>	<b>Outline of the Study .....</b>	<b>4</b>

## CHAPTER 2

<b>2.</b>	<b>LITERATURE STUDY.....</b>	<b>6</b>
<b>2.1</b>	<b>Introduction.....</b>	<b>6</b>
<b>2.2</b>	<b>System Development Methodologies .....</b>	<b>6</b>
2.2.1	Why use a Systems Development Methodology? .....	7
2.2.2	Definition of a System Development Methodology .....	8
2.2.3	Some examples of Formal System Development Methodologies .....	10
<b>2.3</b>	<b>Some Examples of systems development methodologies developed for telecommunications software.....</b>	<b>26</b>
<b>2.4</b>	<b>Comparison of Systems Development Methodologies.....</b>	<b>36</b>
<b>2.5</b>	<b>Mobile Telecommunication Environment.....</b>	<b>46</b>
<b>2.6</b>	<b>Conclusion .....</b>	<b>55</b>

## CHAPTER 3

<b>3.</b>	<b>RESEARCH DESIGN.....</b>	<b>56</b>
<b>3.1</b>	<b>Introduction.....</b>	<b>56</b>
<b>3.2</b>	<b>Research Paradigm .....</b>	<b>56</b>
<b>3.3</b>	<b>Research Approach.....</b>	<b>57</b>
<b>3.4</b>	<b>Research Method .....</b>	<b>59</b>
<b>3.5</b>	<b>Data Collection Method.....</b>	<b>64</b>
<b>3.6</b>	<b>Data Coding .....</b>	<b>74</b>
<b>3.7</b>	<b>Data Analysis.....</b>	<b>74</b>
<b>3.8</b>	<b>Conclusion .....</b>	<b>77</b>

## **CONTENTS (continued)**

### **CHAPTER 4**

<b>4.</b>	<b>RESULTS OF CASE STUDIES AND FINDINGS.....</b>	<b>78</b>
<b>4.1</b>	<b>Introduction.....</b>	<b>78</b>
<b>4.2</b>	<b>Research aims and objectives .....</b>	<b>78</b>
<b>4.3</b>	<b>Results.....</b>	<b>79</b>
4.3.1	Case 1: Company 1 .....	80
4.3.2	Case 2: Company 2 .....	88
4.3.3	Case 3: Company 3 .....	93
<b>4.4</b>	<b>Cross case analysis.....</b>	<b>101</b>
<b>4.5</b>	<b>Cross Case Analysis Findings .....</b>	<b>102</b>
4.5.1	Systems development methodology used .....	102
4.5.2	Key aspects of mobile telecommunication software development.....	110
4.5.3	Identify a list of necessities a systems development methodology should contain, when used to develop mobile telecommunications software.....	114
<b>4.6</b>	<b>Conclusion .....</b>	<b>120</b>

### **CHAPTER 5**

<b>5.</b>	<b>SUMMARY AND FINAL CONCLUSIONS .....</b>	<b>121</b>
<b>5.1</b>	<b>Introduction.....</b>	<b>121</b>
<b>5.2</b>	<b>Research contributions.....</b>	<b>121</b>
<b>5.3</b>	<b>Research aims and objectives .....</b>	<b>123</b>
<b>5.4</b>	<b>Results of the Study .....</b>	<b>123</b>
5.4.1	Systems development methodology used .....	123
5.4.2	Key aspects of mobile telecommunication software development.....	126
5.4.3	A list of necessities a systems development methodology should contain, when used to develop mobile telecommunications software.....	127
<b>5.5</b>	<b>Limitations of the study and future work.....</b>	<b>129</b>

<b>BIBLIOGRAPHY .....</b>	<b>131</b>
---------------------------	------------

<b>APENDIX A .....</b>	<b>140</b>
------------------------	------------

# CHAPTER 1

## 1. INTRODUCTION

### 1.1 Introduction

The technology in the mobile telecommunication industry is constantly changing. Sanders *et al.* (2003) state that “Compared with other industries, the rate of change in mobile technology is unprecedented”. That is because, the higher the volumes, the lower the technology cost, the faster the technology and its supported services evolve (Halonen, Romero, & Melero, 2003). This requires software to be developed at an ever increasing pace. Studies have shown that the proper use of a systems development methodology (SDM) is beneficial to the system development process (See Fitzgerald (1996) for a summary). The problem with telecommunication systems is that they have specific needs that cannot be handled properly by existing software engineering tools (Patel, 2002). There are several properties that distinguish the telecommunications industry from normal software systems:

- The distributed nature of telecommunication services software (Patel, 2002)
- The fact that the size of telecommunication networks tend to grow exponentially (Patel, 2002)
- There is a constant introduction of new services and products (Mehra, 2005)
- Telecommunications systems are extremely complex (Patel, 2002)
- The requirements imposed on telecom systems are becoming increasingly ambitious and complex (Kalayanasundaram *et al.*, 1998)
- There is increasingly less time available to develop these systems (Patel, 2002).

It is the software developer’s task to design, build and implement extremely complex software that utilizes the new services and products. This software has to be developed at an ever increasing rate, while at the same time maintaining quality and minimizing the development cost.

### 1.2 Research aims and objectives

This research aims to determine how mobile telecommunication systems are developed in South Africa by focusing on the use of systems development methodologies. To determine this, the following three main research objectives are proposed:

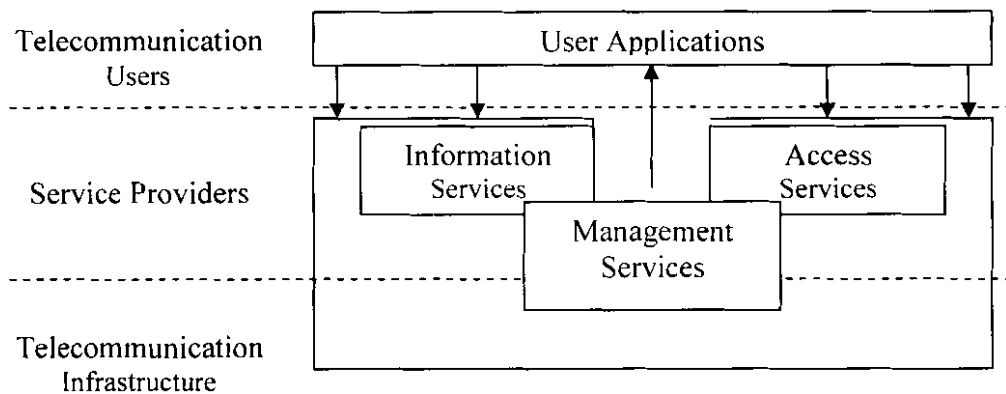
- Identify the systems development methodologies used, if any, by companies in South Africa in the development of telecommunication systems
- To highlight certain key aspects of mobile telecommunication software development
- Identify a list of necessities a systems development methodology should contain when used to develop mobile telecommunications software.

In this study these objectives will be met by first identifying systems development methodologies that are currently available for mobile telecommunication software development. Secondly the above systems development methodologies will be compared according to the framework suggested by Avison and Fitzgerald (2002) in order to determine whether and how they are used during the development of telecommunications systems. Thirdly a list of necessities will be identified that should be present in a systems development methodology when developing telecommunications systems. It will also be determined how the above systems development methodologies adhere to the list. Fourthly a qualitative case study will be done at different organisations which focus on the development of telecommunication systems, and through qualitative methods, determine how telecommunication software development is performed in practice. By doing this the relevance of the above mentioned list of necessities will be tested.

The extreme importance and rapid growth of the mobile telecommunications industry in South Africa makes it necessary to look at how mobile telecommunication systems are being developed. In this highly competitive market it is extremely important that mobile telecommunications systems are developed in minimal time with minimal cost while still ensuring quality and usability. A big factor that could have a positive impact on the development process is the use of a systems development methodology (Fitzgerald, 1996). The results of this study will enable mobile telecommunication software developers to make better decisions regarding the use of a systems development methodology. It will also facilitate the choice of an appropriate systems development methodology, part of a systems development methodology or creation of an in-house systems development methodology.

### 1.3 Telecommunications Systems

It should be noted that there are many different types of systems in the telecommunications industry. The different types of systems are displayed and explained in the figure below (Fig. 1.1).



**Fig.1.1 Service model of telecommunication systems (Patel, 2002).**

Figure 1.1 represents a structured model for telecommunication software (Patel, 2002). The definition of the model is given as in Patel (2002). At the bottom is the *Telecommunication Infrastructure* layer. The main function of this layer is the transmission of data through telecommunication networks. In the middle is the *Service Provider* layer. A service provider is a link between user applications and the telecommunications infrastructure. The top layer is a *Telecommunication Users* layer consisting of the user applications. There are several services contained within the model. *Information Services* (offered by the service provider layer) offers network information services, providing users with high level information, information hosting and delivery services which support users in the delivery of information over the network. *Access Services* (offered by the service provider layer) gives users access to data transmission and management functions. *Management Services* are concerned with the management of telecommunication hardware and software. They control network and application management and user response services.

In this research the focus will not be on a specific level of the service model, but a generalized view of the telecommunications industry as a whole.



## **1.4 Qualitative research**

This research will gather information on systems development methodologies used in practice by means of conducting qualitative case studies at telecommunication software development companies. Firstly a study of qualitative case study methods will be done. This information will be used in developing a suitable list of questions to be used during the semi-structured interviews as well as determining how the interviews should be conducted. The factors pertaining to the case study will be to draw up a suitable set of questions, perform an in-depth interview with a developer or/and project manager and gather any information from available documentation. The data collected will then be analyzed and used to create propositions regarding the objectives of the study. During the analysis of the case studies, the interviewees were contacted whenever there was information lacking or issues that needed further discussion. The final propositions were then sent to all the interviewees that took part in the case study to confirm whether the statements were correct.

## **1.5 Outline of the study**

### **Chapter 1: Introduction**

This chapter defines the research aims and objectives of the study.

### **Chapter 2: Literature study**

In this chapter the available literature on systems development methodologies and the mobile telecommunication industry is reviewed. This chapter contains a definition of a systems development methodology as well as examples of formal systems development methodologies. Further three systems development methodologies developed for use in the mobile telecommunications domain are discussed. A list of software development needs that must be adhered to when developing a mobile telecommunication system will be given.

### **Chapter 3: Research design**

This study is based on qualitative research and in this chapter the qualitative research method will be discussed. The qualitative research method used during the study will be discussed.

### **Chapter 4: Results of case studies and findings**

The results of the case studies are presented as well as findings based on the analysis of the data. The results focus on whether a systems development methodology was used, followed by a

description of the systems development methodology. Thereafter some key elements of telecommunication software development are identified and discussed. Lastly the list of software development needs that should be adhered to when developing a mobile telecommunication system will be reviewed and revised using the case study results.

## **Chapter 5: Conclusion**

This chapter contains a summary of the results. Any shortcomings of the study are also noted.

## **CHAPTER 2**

### **2. LITERATURE STUDY**

#### **2.1 Introduction**

In this chapter the available literature on the subject of systems development methodologies and the mobile telecommunication industry will be reviewed.

Firstly a discussion of some of the arguments for and against using a systems development methodology during development will be given. Thereafter a definition of a system development methodology is given and some examples of formal (well used and respected) systems development methodologies are given. The definition of “formal” in this context is the systems development methodologies are well used and respected.

The second part will view a few of the systems development methodologies that were designed to be used in the telecommunications industry. These systems development methodologies will then be compared by means of the framework of Avison and Fitzgerald (2002). The aim of this comparison will be to see whether there are any differences or similarities between the systems development methodologies specifically designed for the telecommunications industry. Some insight may be gained into the characteristics of the systems development methodologies and also whether the systems development methodologies are used in practice.

The last part of the chapter will contain a short discussion of the telecommunications environment. The focus will be on the characteristics of the telecommunications industry, as well as any specific challenges and needs when developing mobile telecommunication systems.

#### **2.2 System Development Methodologies**

A part of the research aims and objectives is to determine whether any formal systems development methodologies are suitable to use in telecommunications software development. Firstly a discussion of some of the advantages as well as disadvantages of using a systems development methodology during the development process is given. Following this will be the definition of a systems development methodology as it will be used in this study. Thereafter some examples of different formal systems development methodologies will be discussed briefly.

### 2.2.1 Why use a Systems Development methodology?

As the popularity of computer systems increased during the last few decades, so did the need for better software development. The *old* way in which the system developers were technically trained but lacked good communication skills was no longer adequate. This led to systems that did not meet the users' needs and were completed, more often than not, late and over budget.

Much of the developer's time was spent on correcting and enhancing the applications that were operational (Avison & Fitzgerald, 2002). Because of these inherent problems to system development, the term "software crisis" was coined.

It was clear that a change in the approach to systems development was required. During the 1968 NATO Software Engineering Conference held in Germany, the conclusion of the conference participants was that software engineering should use philosophies and paradigms similar to those used in other engineering disciplines to solve the software crisis.

Numerous systems development methodologies have since been developed, giving rise to much debate about the usefulness of systems development methodologies. Some of the proposed advantages and disadvantages are:

- **Advantages**

Some of the advantages of systems development methodologies are that they help in **controlling** the development process. This added control leads to other improvements such as, faster development speed and lower development cost (De Vries, 2004).

The use of a systems development methodology also has an effect on the **quality** of the developed product, ensuring the product quality by prescribing a framework with measurements and criteria for their execution (Purvis & Sambamurthy, 1997; Avison & Fitzgerald, 2002).

Because of the common approach to development that a systems development methodology provides, the way in which projects are designed, developed and implemented can be **standardised**. Having a standardised approach to systems development means that management and control of the project are improved, while productivity is increased. Some systems development methodologies focus on developing components that can be reused in different projects (Avison & Fitzgerald, 2002).

By using a systems development methodology, any knowledge acquired during the development process can easily be stored and can serve as a **learning tool** in later projects (Glasser, 1998).

- **Disadvantages**

De Vries (2004) notes that systems development methodologies tend to try and **generalize** the development process and do not take any variation into account.

Systems development methodologies tend to be overly **complex** and require much **skill** to use. This is because of the great depth into which the systems development methodology goes to describe the different tasks that should be followed during development (Avison & Fitzgerald, 2002). In turn, this could lead to much **resistance when adopting** the systems development methodology.

**Productivity** could suffer when a systems development methodology is used, because more time has to be spent on completing task and documentation (Avison & Fitzgerald, 2002). According to Fitzgerald (1996), systems development methodologies are **unsuitable for current day systems** because of the rapid changes in the IT industry and the increased speed at which systems have to be developed.

Some that have used systems development methodologies found that, using them does not give better results, and in some cases they have actually hindered project success (Avison & Fitzgerald, 2002).

### **2.2.2 Definition of a System Development Methodology**

There has been much of discussion about a universally accepted definition of a systems development methodology. A few of the definitions are listed below:

- The British Computer Society Information Systems Analysis and Design Working Group defines a systems development methodology as a recommended collection of philosophies, phases, procedures, rules, techniques, tools, documentation, management and training for developers of information systems (Maddison *et al.*, 1983).
- A collection of procedures, techniques, tools and documentation aids which will help the system developers in their effort to implement the new information system. A systems development methodology will consist of phases, from the feasibility study, through to maintenance, which themselves can consist of sub phases, guiding the system developers in

their choice of the techniques that might be appropriate at each stage of the project and also help them plan, manage, control and evaluate information systems projects. A systems development methodology also has a philosophy (Avison & Fitzgerald, 2002).

- A system development methodology comprises an overall strategy for computer-based information systems development that includes a flexible framework of the sequence of development tasks along with the techniques used to accomplish each task (Roberts *et al.*, 1998).
- A systems development methodology is a systematic procedure for completing either a system or one of several stages of the systems development life cycle. It consists of goals, principles, specific methods and tools, which are selected on the basis of an underlying rationale or systems development philosophy (Iivari, Hirschheim, & Klein, 2000).

Generally it seems that a systems development methodology is the steps developers take to develop a system. Not only the steps, but also the order in which those steps are taken as well as the activities performed in each step. A systems development methodology is more than just this. As stressed by Avison and Fitzgerald (2002), a systems development methodology is also based on some philosophical view.

In a sense one can define a systems development methodology as a combination of the following four elements (Huisman & Iivari, 2006):

- **Approach(es)**  
This is the philosophy on which the systems development methodology is built. Different approaches could be concerned with cost, quality documentation, development speed, adaptability or efficiency. Examples of system development approaches are the structured approach, object-oriented approach, information modeling.
- **Process model(s)**  
Every model defines a certain order or sequence of stages through which a system evolves (Wynekoop & Russo, 1993). Examples are the linear life cycle model and the Boehm's spiral model.

- **Method(s)**

A method is a systematic approach to conducting at least one complete phase of system development, consisting of a set of guidelines, activities, techniques and tools, based on a particular philosophy of system development and the target system (Wynekoop & Russo, 1993).

Examples are OMT, IE.

- **Technique(s)**

The procedure used to perform a development activity (Brinkkemper, 1996). An example of a technique is the entity relationship diagram.

### **2.2.3 Some Examples of Formal Systems Development Methodologies**

In this section, short discussions of some of the formal systems development methodologies that are in use today will be given. The reason for the discussion is to demonstrate some of the systems development methodologies that are available for systems development and to show the spread of different approaches, process models, methods and techniques.

The discussion of the systems development methodologies will be done as follows: As in Avison and Fitzgerald (2002), the systems development methodologies are divided into categories. The category to which each systems development methodology is assigned was developed by Avison and Fitzgerald (2002) and is loosely based on the approach followed by the systems development methodology. First two process-oriented systems development methodologies will be discussed, namely STRADIS and YSM. Secondly the blended systems development methodologies SSADM and Information Engineering will be discussed. Thirdly the object-oriented systems development methodologies OOA and RUP will be discussed. Fourthly a discussion of the rapid systems development methodologies JMRAD, DSDM and XP will be given. The fifth systems development methodology that will be discussed is the people-oriented ETHICS systems development methodology and lastly, the organisational-oriented SSM. The way the systems development methodologies are discussed will be according to the four elements of the definition given in 2.2.2.

### **2.2.3.1 Process-oriented system development methodologies**

The system development methodologies that fall into this category make use of process-oriented techniques. Examples of these techniques are functional decomposition, data flow diagrams, decision trees, decision tables and structured English (Avison & Fitzgerald, 2002). The system development methodologies that will be discussed here are STRADIS and YSM.

- **STRADIS**

The structured analysis, design and implementation of information systems (STRADIS) methodology for developing systems is a structured systems development methodology that focuses on the use of techniques. The following discussion of the technique comes from Gane and Sarson (1979).

#### **Approach**

STRADIS follows a top-down, process-oriented approach.

#### **Process Model**

The STRADIS systems development methodology describes a series of phases which are followed in a linear fashion, while the physical design phase has some activities that should be run parallel.

#### **Method**

##### **4 Phases:**

1. **Initial study**

The first step is to gather information from users and documents to determine whether building the system justifies the monetary cost. An overview data flow diagram of the proposed system is then constructed.

2. **Detailed study**

Identify and interview the system users to ascertain their requirements and interests. These requirements are then used to construct a logical model of the current system.

3. **The third step involves the definition and design of alternative solutions. To determine these alternative solutions the system objectives are reviewed and a logical data flow diagram of the system is produced.**

Three different design alternatives are then setup: Low-budget, mid-budget and high-budget.



#### 4. Physical design

After choosing the optimal alternative, this alternative is then refined. During this refinement process the file structure, modular hierarchy and clerical tasks are derived from the DFD.

#### **Techniques**

The system is graphically modeled via the Data Flow Diagram (DFD) which is stored in a data dictionary. Other techniques used according to Avison and Fitzgerald (2002) are decision trees, decision tables, structured English, structure diagrams, action diagrams, entity life cycles and matrices.

#### **Characteristics**

Although the systems development methodology can be used in the development of any type of information system whether big or small in size, STRADIS is most relevant to a situation in which there is a backlog of systems waiting to be developed and insufficient resources to devote to all potential new systems. It also incorporates a wide variety of established techniques.

- **Yourdon Systems Method (YSM)**

The Yourdon Systems Method as described in Yourdon Inc. (1993).

#### **Approach**

YSM follows a process-oriented approach which is not entirely top-down nor bottom-up, but more middle-out.

#### **Process Model**

This systems development methodology, like STRADIS, also follows a linear type process model.

#### **Method**

The three main phases are:

1. Feasibility study

Here the analyst looks at the current system and tries to establish what the system does. During this study, an overview dataflow diagram is drawn up.

## 2. Essential modeling

Within the essential modeling phase, two models are constructed, these being the environmental model and the behavioural model. The environmental model describes how the system fits into the surrounding environment. The behavioural environment defines how the system must behave to deal successfully with the environment. Within both models extensive use is made of data flow diagrams.

## 3. Implementation modeling

This is the first step in designing the system. The models created so far are examined to determine boundaries of computation and determine how data might be stored

### **Techniques**

The system is graphically modeled via the Data Flow Diagram (DFD) which is stored in a data dictionary. Other techniques used according to Avison and Fitzgerald (2002) are decision trees, decision tables, structured English, structure diagrams, action diagrams, entity life cycles and matrices.

### **Characteristics**

Originally YSM was similar to STRADIS. However, in recent versions of the systems development methodology the approach has changed to a 'middle-up' approach. The systems development methodology focuses on the analysis of the processes as well as the data. In addition to covering the systems development aspect, YSM also covers the organisational activities.

#### **2.2.3.2 Blended Systems Development Methodologies**

Blended system development methodologies utilize a blend of process and data oriented techniques. The system development methodologies that will be discussed are SSADM and IE.

- **SSADM**

The structured systems analysis and design method (SSADM) is a highly structured systems development methodology. The description below is based on Weaver, Lambrou and Walkley (1998).

### **Approach**

This systems development methodology follows a “blended” approach emphasizing processes as well as data elements of the system.

## **Process Model**

SSADM adopts the Waterfall model of systems development, where each phase has to be completed and signed off before subsequent phases can begin.

## **Method**

SSADM consists of five modules each with one or more stages. These modules are:

1. Feasibility study

### **Stage 1 – Feasibility**

Firstly the scope of the project is assessed. A feasibility study is then done to determine the best system design option from a number of alternatives.

2. Requirements analysis

### **Stage 2 – Investigation of current environment**

This step expands the detail of the scope assessment done in the previous stage, resolving any issues that may exist in the current models.

### **Stage 3 – Business system options**

Some business systems options that satisfy the minimum set of user requirements are outlined. One of these is then chosen by management and forms the basis of the system specification.

3. Requirements specification

### **Stage 4 – Definition of requirements**

Here the full requirements specification is determined and documented. Optionally, prototypes can be developed demonstrating critical dialogs and menu structures.

4. Logical Systems Specification

The two stages in this phase are carried out parallel.

### **Stage 5 – Technical system options**

The aim of this stage is to determine the system environment in terms of “hardware and software configuration, development strategy, organisational impact and system functionality”.

### **Stage 6 – Logical design**

The logical design of the system states what the system should do. Other tasks performed in this stage include defining the dialog and menu structures as well as the system processes and process sequences.

## **5. Physical Design**

### **Stage 7 – Physical design**

Prepare, optimize and assemble the physical design.

The stages following design are not covered by the systems development methodology because they are seen as being “installation-specific”.

### **Techniques**

The two main techniques used in the systems development methodology are the process techniques entity modeling and normalization. Logical Data Modeling and Dataflow Modeling are also done.

### **Characteristics**

SSADM is best used when developing large projects that have specific and measurable requirements. The systems development methodology is data-driven, focusing on data modeling and database design.

Thanks to the specific, time-bound goals and deliverables inherent to the systems development methodology, there is a considerable improvement in project structure, planning and management. However, the systems development methodology is difficult to learn and apply, and its rigid approach discourages user adoption and creativity (Middleton, 2000).

- **Information Engineering (IE)**

This description of Information Engineering is based on the work of Martin (1989).

### **Approach**

This systems development methodology follows a “blended” approach emphasizing processes as well as data.

### **Process Model**

The IE systems development methodology makes use of the classic waterfall type model for developing a system.

## **Method**

The IE systems development methodology includes four main levels or layers: (Avison & Fitzgerald, 2002)

1.    **Information strategy planning**  
      The aim here is to construct an information architecture and strategy for the entire organisation to identify relevant business areas.
2.    **Business area analysis**  
      The business area's system requirements are then mapped out.
3.    **System planning and design**  
      Establish what the user wants and how it can be attained using technology.
4.    **Construction and cutover**  
      Build and implement the system laid out in previous levels 1-3.

## **Techniques**

According to Avison and Fitzgerald (2002), the two main techniques used in the systems development methodology are the process techniques entity modeling and normalization. Besides that, the main documents used in the systems development methodology are data-oriented entity relationship diagrams, decision trees, data flow diagrams and dependency diagrams (Laurido-Santos, 1986).

## **Characteristics**

Information Engineering is a data centered approach. Data centered means that the development process has more emphasis on data, although some process-oriented aspects are embedded in the systems development methodology (Avison & Fitzgerald, 2002). A drawback of the IE systems development methodology is that the strategic planning phase is open ended and this increases the possibility of this phase never being completed.

### **2.2.3.3      Object-oriented Systems Development Methodologies**

Here the systems development methodologies that will be discussed are Object-Oriented Analysis as well as RUP.

- **Object-Oriented Analysis (OOA)**

This description of Object-Oriented Analysis is based on the pre-UML (unified modeling language) version of the systems development methodology, as can be found in Coad and Yourdon (1991).

## **Approach**

This systems development methodology follows an object-oriented approach.

## **Process Model**

Although the systems development methodology is sometimes described as a linear set of activities, it is also possible to iterate the process.

## **Method**

The OOA systems development methodology consists of five major activities:

1. Finding Classes and Objects

This activity increases the analyst's understanding of the problem domain. Some of the methods used to gather this information are: first-hand observation, interviews with "domain experts", reading and viewing related systems and prototyping.

2. Identifying Structures

Here the classes and objects are organised into hierarchies, which enable class inheritance.

3. Identifying Subjects

This activity divides the model into more manageable groups. This helps to reduce the complexity of the model produced so far.

4. Defining Attributes

This activity defines the data elements of the objects, while giving more attention to attributes that define the state of the object.

5. Defining Services

While the previous step defined the data, this step defines the processing or services. A service, according to Avison and Fitzgerald (2002), is the operation or processes performed by the object in response to the receipt of a message.

## **Techniques**

Object orientation and UML are the main techniques used in this systems development methodology.

## **Characteristics**

The OOA systems development methodology encapsulates data and procedures into groups called objects or classes. These classes intercommunicate by sending messages to one another. An important feature of OOA is the activity of identifying reusable objects, classes and services.

- **Rational Unified Process (RUP)**

This description of the Rational Unified Process is taken from Kruchten (2000).

## **Approach**

The systems development methodology follows an Object-Oriented approach.

## **Process Model**

RUP is an iterative and incremental approach for developing object-oriented systems.

## **Method**

RUP has a number of 'cycles' that make up the development of a project and run throughout its life.

Each of these is made up of four phases:

1. Inception  
Here the focus is on understanding the project scope.
2. Elaboration  
This phase includes understanding the architecture and the requirements of the project.
3. Construction  
Construction of the software.

4. Transition  
The software is released and deployed (West, 2002).

Kruchten (2000) states that during each of the four phases, the RUP goes through a series of nine workflows; a workflow being a sequence of activities that produce a result of observable value.

These nine workflows are:

1. The business modeling workflow
2. The requirements workflow
3. The analysis and design workflow
4. The implementation workflow

5. The test workflow
6. The development workflow
7. The configuration and change management workflow
8. The project management workflow
9. The environment workflow

The reason for these iterations is to minimize risk and deliver executable software after each iteration. These deliverables should be demonstrable and testable against the requirements of the project and the use cases.

### **Techniques**

Object orientation and UML are the main techniques used in the systems development methodology. Extensive use is made of use case tools and other artifacts.

### **Characteristics**

The systems development methodology focuses on managing the risks and delivering results, as well as preventing the development team from procrastinating on activities (Knutson, 2003). The architecture that contains details on the operating system, hardware, the database, networks, etc., is outlined at the beginning, but evolves and develops in tandem with the needs of the software system as it develops (Kruchten, 2000). In many cases, RUP has to be configured to suit users' needs before it is implemented.

It tends to improve projects with iterative processes that are exceeding their scheduled deadlines. RUP is not generally considered an agile approach to development as it contains extensive guidelines for process phases. However, it can be adjusted to be more agile, and the key to this lies in the adoption phase (Abrahamsson, Salo, Ronkainen, & Warsta, 2002).

#### **2.2.3.4 Rapid development Systems Development Methodologies**

The main use of rapid development systems development methodologies is to develop usable software more quickly and provide timely and regular visibility of the solution to customers, product owners and other key stakeholders. A characteristic of rapid development systems development methodologies is that the system being developed is released as a series of iterative parts, each building on the one before.

The following system development methodologies will be discussed: James Martin's RAD, DSDM and XP.



- **James Martin's RAD (JMRAD)**

This discussion of James Martin's Rapid Application Development is based on the work of Martin (1991).

### **Approach**

The main approach for this systems development methodology is based on speed of development.

### **Process Model**

JMRAD can be described as having an evolutionary development process in which it releases the system in 90 day cycles.

### **Method**

JMRAD has four phases:

1. Requirements planning  
Much effort is put into the definition of requirements phase. Here the high-level management requirements of the system are identified using the "joint requirements planning" and "joint application design" (JAD) techniques.
2. User design  
JAD is the main technique used in the design phase. The emphasis is on getting the requirements as correct as possible and reflecting the business needs.
3. Construction  
This phase is undertaken using a toolset. Prototypes are designed for each transaction which the user must approve. After the user's approval the construction of the system is performed by a small group of experts, using a toolset.
4. Cutover  
This phase involves testing the system using realistic data. The old and the new system are run in parallel, until the new system has proven itself.

### **Techniques**

JMRAD relies on the techniques of stakeholder analysis and joint application design. It also emphasizes the use of project management tools and techniques during development.

## **Characteristics**

JMRAD is a systems development methodology that has been designed with the increasing competitiveness of today's business environment in mind. It speeds up the development process which is becoming increasingly necessary because of changing business needs. JMRAD adopts an evolutionary/prototyping approach. It focuses on identifying and involving important users and obtaining commitment from them. The systems development methodology also makes extensive use of tools and requires a sophisticated repository.

- **Dynamic System Development Method (DSDM)**

The Dynamic Systems Development Method is based on JMRAD and was developed partly with the help of British telecom. A detailed definition of DSDM version 4.2 can be found at the DSDM consortium website (DSDM, 2003).

## **Approach**

The main approach for this systems development methodology is speed of development.

## **Process Model**

The process follows an iterative approach, as well as RAD style timeboxing.

## **Method**

There are five main phases in the development life cycle:

1. **Feasibility study**  
The feasibility study is a short, sharp exercise. Howard (1997) states that DSDM is not recommended for real time applications.
2. **Business study**  
This phase is to gain understanding of the business processes involved.
3. **Functional model iteration**  
Here the requirements gathered from the business study are refined, followed by the development of prototypes and software.
4. **System design and build iteration**  
This is where the system is built, ready for delivery to the users.

## 5. Implementation

This is the cutover from the existing system to the new one, as well as the completion of the user's manual and other documentation.

### **Techniques**

DSDM is much like JMRAD discussed previously as it also relies on the stakeholder analysis and joint application development techniques. Furthermore it emphasizes the use of project management tools and techniques during development.

### **Characteristics**

DSDM provides a framework of controls for building and maintaining systems which meet tight time constraints. User involvement plays an important role in system design and development.

- **Extreme Programming (XP)**

This discussion of Extreme Programming is based on the work of Jeffries (2001).

### **Approach**

The main approach for this systems development methodology is speed of development.

### **Process Model**

The development follows an iterative process.

### **Method**

XP was created to handle projects with changing requirements where the functionality of the system is expected to change often.

XP consists mainly of rules and practices for making systems development faster. When used together, a systems development methodology emerges consisting of four phases:

#### 1. Planning

“User stories” are extensively used to define the system requirements and scope. This stage also identifies the team members, contents of each increment and the estimated system cost.

#### 2. Design

Architectural spikes or prototypes are used to create a simple overall.

CRC Cards encourage all team members to understand and contribute to the system design.

XP also relies on a programming technique called re-factoring to help uncover the most effective system architecture.

3.     **Developing the code**

This phase is all about pair programming, re-factoring, and creating tests before the code. This increases the quality of the code produced.

4.     **Testing**

Unit tests and acceptance tests.

**Techniques**

Some of the techniques that are used when developing a system using the XP systems development methodology are test driven development, pair programming and continuous integration. During the design phase, a technique known as refactoring is used.

**Characteristics**

When following the extreme programming systems development methodology, the system engineering group consists of a small team of developers (3 to 10), working closely with customers, preferably at one location.

XP improves a software project in four essential ways: communication, simplicity, feedback, and courage (Extreme Programming, 2006).

XP has the advantage of being a “light weight” systems development methodology, which means that there are less stringent rules to learn and follow. The main aim is delivering the software and not on other deliverables such as documentation. The iterative approach of the systems development methodology, together with code testing, results in a lower cost, better quality system.

A disadvantage of XP is that the development team should be close together to allow for adequate communication.

**2.2.3.5     People-oriented Systems Development Methodologies**

•     **ETHICS**

The discussion of the Effective Technical and Human Implementation of Computer-base Systems (ETHICS) is based on the work of Mumford (1995).

**Approach**

ETHICS is a people-oriented approach. It gives great importance to the job satisfaction of the development team, also user involvement and participation during the design of the system.

### **Process Model**

The systems development methodology follows a linear process, but also suggests that some steps be done in parallel.

### **Method**

ETHICS has 15 steps:

- Step 1: Why change?
- Step 2: System boundaries
- Step 3: Description of existing system
- Steps 4, 5 & 6: Define the key objectives and tasks
- Step 7: Diagnosis of efficient needs
- Step 8: Diagnosis of job satisfaction needs  
(Uses a general questionnaire to measure job satisfaction needs)
- Step 9: Future analysis
- Step 10: Specifying and weighting efficiency and job satisfaction needs and objectives
- Step 11: The organisational design of the new system
- Step 12: Technical options
- Step 13: The preparation of a detailed work design
- Step 14: Implementation
- Step 15: Evaluation

The steps that should be done in parallel are steps 11 and 12.

### **Techniques**

Stakeholder analysis is the main technique used in this systems development methodology.

### **Characteristics**

ETHICS has a socio-technical approach that takes into account the interaction of technology and people. The systems development methodology focuses on the issue of job satisfaction and user participation.

### **2.2.3.6 Organisational-oriented System Development Methodologies**

- **Soft Systems Methodology**

This discussion of the Soft Systems Methodology is based on the work of Checkland and Scholes (1999).

#### **Approach**

The systems development methodology offers an organisational-oriented approach which does not only look at the system, but also the system of which it is a part.

#### **Process Model**

When going through the seven stages of the systems development methodology, some stages could be done simultaneously, also, iteration and backtracking are essential.

#### **Method**

The SSM approach is iterative. The systems development methodology is set in seven stages:

1. Unstructured problem situation
2. Problem situation expressed
3. Root definition expressed
4. Conceptual model
5. Real world/ conceptual model comparison
6. Feasible and desirable change
7. Action to improve the situation.

#### **Techniques**

This systems development methodology does not specify any specific techniques or tools that should be used during the development process.

#### **Characteristics**

The soft systems methodology gives an answer to the inadequacies of structured systems development methodologies. It takes into account the social component of systems development.

The main focus is on problem identification and resolution rather than a solution.

## **2.3 Some examples of systems development methodologies developed for telecommunications software**

In the above section some formal systems development methodologies that are available for use when developing a computerized system are discussed. Unfortunately they are not always well suited for application development in the mobile industry (Holler, 2006; Patel, 2002; Vallecillo, 2001). When doing a study on mobile telecom specific systems development methodologies the literature was limited. Due to this lack of available information it was decided to also use systems development methodologies focusing on normal telecommunication system development and in so doing, represent systems development within the entire telecommunications environment.

Three systems development methodologies are discussed below. Firstly the MODA-TEL systems development methodology will be discussed as described in Steinhau (2003). After this the Mobile-D development systems development methodology will be discussed (VTT, 2006) followed by a systems development methodology discussed in Mansurov (2000) that utilizes formal methods in the accelerated development of telecommunication software. The systems development methodologies will be discussed according to the four elements of the definition given in 2.2.2.

### **2.3.1 MODA-TEL**

MODA-TEL (Steinhau, 2003) was developed specifically for the development of telecommunication management networks in a joint effort by European telecommunication network operators and service providers. The aim of MODA-TEL is to define some key systems development methodologies for applying MDA (Model Driven Architecture) to the development of telecommunication projects. This definition, however, is not meant to be a rigid definition, rather, it is meant to give guidance to developers in modeling and model transformation (Steinhau, 2003).

#### **Approach**

The MODA-TEL systems development methodology follows an object-oriented approach because it is based on the Model Driven Architecture which has certain object-oriented characteristics.

#### **Process Model**

The main phases discussed in the systems development methodology as well as the phases followed during the development process all follow the object-oriented style iterative process model.

## Method

The authors of the paper discuss the need to separate between “preparation activities and execution activities”. It should also be possible to switch between the two as needed. The main phases of MODA-TEL, as discussed in Gavras, Belaunde, Pires, and Almeida (2004) are designed to create a knowledge base which can be reused in later projects.

The following phases are identified: (Gavras *et al.*, 2004)

1. Project management

Organising and monitoring the project are the main concerns. First an appropriate Software Development Process (SDP) is identified, which will be used in the project execution phase. Because MDA is based on object-orientation principals, most existing development processes can be used. After the identification of the SDP, the assignments of activities to specific roles are determined. Any procedures for enhancing the quality of the product are also identified. Project management in this context does not refer to “keeping track of milestones and resource consumption” (Gavras *et al.*, 2004, p. 73), but rather focuses on “the management decisions absolutely necessary to setup the project” (Gavras *et al.*, 2004, p. 73). The project management phase spans the entire life of the project.

Steps two, three and four are defined as being the project preparation activities. Although they are setup before project execution they are often revisited and revised throughout the project development process.

2. Preliminary preparation

Here the modeling and transformation needs of the project are identified (i.e. Where in the project will they be required?).

There are four activities in this phase. First: Selecting the platform to be used. Second: Identifying the modeling language that will be used in representing data, business process specification, user requirements etc. Third: Defining a way of transforming the model information of the abstract platform to the concrete platforms. Lastly: To provide traceability to keep track of these changes.

3. Detailed preparation

In this phase the needs, roles and annotations identified in the preliminary preparation are used to setup the specification of the modeling languages as well as the transformations.



4. Infrastructure setup

This step identifies the tools that will be used during development and the management of metadata.

5. Project execution

After the preparation phases, the final product and artifacts can now be produced by means of the SDP chosen in the project management phase. The project execution phase is considered the main phase of the project (Gavras *et al.*, 2004) and the activities followed are specific to the SDP chosen in the project management phase.

Because of the MDA based approach of the systems development methodology, Steinhau (2003) does discuss some of the key MDA principals that should be followed during the phases of the selected SDP (Steinhau, 2003).

- Requirements phase

During the requirements phase, early information models are created. The terminology of these models should be well defined and documented because they could later be used in the creation of design-oriented models and test specifications.

- Analysis and design phase

When analyzing the existing system, Steinhau (2003) recommends using domain specific modeling which extracts the domain's meta-models or profiles. From these, models are then created to describe the system formally.

- Implementation phase

After most of the required information has been modeled, the implementation is much more structured. One of the steps here is to create or improve the algorithms used in model transformation or code generators.

- Test and integration phase

Test cases should be well modeled, utilizing industry standards like TTCN3 (Testing and Test Control Notation).

- Deployment phase

The deployment process can be made much more transparent by using model transformation to create the meta-models and profiles used for deployment information.

- Maintenance phase

The MDA advantages of traceability and model references could be a great help during system maintenance. The maintenance of a system can be made much cheaper and more effective by architecting the system in a dynamic and structured manner.

### **Techniques**

The modeling language and techniques used during the project execution stage are dependent on the system development process selected during phase one. In Steinhau (2003) it is recommended that object-oriented systems development methodology like RUP is used, and that UML is used as modeling language.

### **Characteristics**

MODA-TEL is an MDA based systems development methodology which was specifically created for distributed systems. Because MDA is technology independent, the scalability of the product produced by using this systems development methodology as well as the flexibility is improved. The object-oriented nature of the systems development methodology enables the reuse of existing development processes in organisations and projects.

## **2.3.2 Mobile-D**

### **Approach**

The Mobile-D systems development methodology is a type of blended systems development methodology in that it contains characteristics of Extreme Programming (practices), Crystal systems development methodologies (scalability) and Rational Unified Process (coverage) (VTT, 2006). The systems development methodology was specifically created with agility in mind (Abrahamsson, *et al.*, 1994), but it also focuses on speed and object-oriented development.

### **Process Model**

The development process of this systems development methodology follows an iterative model.

### **Method**

The systems development methodology consists of 5 phases, each with several stages and with one or more tasks within each stage (VTT, 2006).

The systems development methodology is ordered into phases as follows:

1. Explore

In this phase the main stages are to establish the stakeholders, defining the scope through requirements collection and establishing the project. In the scope definition the “initial requirements are defined and agreed for the project” and the span of the iterations are determined. These initial requirements are documented in a suitable, agreed to format. A plan is also developed for handling system expansion and growth. Project establishment includes personnel allocation, environment and architecture selection, as well as establishing the process to be followed. This ensures that everything is in place to start development immediately.

2. Initialize

This phase is used to setup the project through setting up the chosen environment and establishing the communications with the customer. Also done in this phase are architecture line planning and initial requirements analysis. Optionally the project team can solve some critical issues without producing any working code.

From here on all the phases work in iterations and have planning, working and release stages.

3. Productionize

The productionize phase has three stages, these are the planning stage, the “working day” stage and the “release day” stage. To develop an entire system, the systems development methodology iterates through these three stages several times.

- Planning

In the planning stage the work content for the iteration is selected by analyzing and prioritizing the user requirements. Acceptance tests are then developed to verify customer requirements. These acceptance tests also serve as a tool for communicating the customer’s needs to the developers. In addition to the acceptance tests, the requirements are also documented by means of developer notes and user interface illustrations. After the planning stage comes a series of “working days” to develop the system.

- **Working day**

In this stage the functionality that was planned in the previous stage is implemented. Here the focus is on “Test driven development”, “Pair programming”, “Continuous integration” and “Refactoring”.

After each day’s work, a “wrap up” session is held to communicate progress as well as any problems within the team and to the customers. After the “working day” stage has gone through enough iterations and has developed a functional part of the system, development continues to the “release day” stage.

- **Release day**

After several “working days” in the iteration, the code developed is tested and approved by the customer and released. One of the major goals of this stage is to integrate subsystems into a single product. Using a version control tool makes this operation much easier. If the system requires further development, the process returns to the planning stage for another iteration.

4. **Stabilize**

The project now goes through a mini iteration of planning, developing and releasing to stabilize the system and wrap-up the documentation.

5. **System Test and Fix**

By doing system and acceptance tests, defects in the software are identified and the defect information is documented for the fix iteration. Fixes are implemented through iterations as in the productionize phase.

### **Techniques**

Project management techniques are used from the explore phase right through the development process. During development, agile techniques like test-driven development, pair programming and continuous integration are used. Other techniques include project retrospective techniques during post-iteration workshops

### **Characteristics**

Mobile-D, as the name states is an agile systems development methodology. The systems development methodology is founded upon the following nine principal elements:

- **Phasing and pacing**  
The projects are performed in iterations of which each begins with a Planning Day.
- **Architecture Line**  
Architecture line approach is utilized together with architectural patterns and Agile Modeling.
- **Mobile Test Driven Development**  
Test-first approach is utilized together with automated test cases.
- **Continuous Integration**  
Effective software configuration management (SCM) practices are applied through multiple means.
- **Pair Programming**  
Coding, testing and refactoring are carried out in pairs.
- **Metrics**  
Few essential metrics are collected rigorously and utilized for feedback and process improvement purposes.
- **Agile Software Process Improvement**  
Post-Iteration workshops are used to improve the development process continuously.
- **Off-Site Customer**  
Customer participates in Planning and Release Days.
- **User-Centered Focus**  
Emphasis is placed on identifying and fulfilling end-user needs.



- c. Create validation scenarios
  - d. Run validation scenarios through the requirements model.
  - e. Validate the execution sequence of each validation scenario to either
    - Accept the validation scenario. In this case the validation scenario can be included into requirements use-cases
    - Reject the validation scenario. In this case the initial customer requirements contain a fault. E.g. the initial requirements can be inconsistent or incomplete. The rejected validation scenario has to be transformed into a use-case and the initial requirements need to be updated by including the new use-case and removing any existing inconsistencies.
  - f. Check termination criteria and start with a new iteration, if necessary (from step 2).
3. Synthesis of SDL requirements models from approved MSC scenarios
- For the process of converting the use-cases to SDL requirements models, the use-cases must first be formalized. Formalizing the use-cases is done by means of the Message Sequence Charts (MSC) language, which takes into account the control flow between scenarios, as well as the data flows. One of the many available tools is then used to translate the MSC models into SDL specifications. (Examples of such tools are: Concordia University synthesizer and the Moscow synthesizer). While looking at related studies, the systems development methodology also mentions generating prototypes, although this is not a requirement of the systems development methodology.
4. Seamless refinements of SDL requirement models into design models
- The first step in this stage is to define the system architecture and to produce “system scenarios” for each architecture component.
- Now the synthesized architecture model (SAM) is created. This model takes into account the architecture, functionality, interactions between architecture components as well as data flows. In this step, an effort is still made to detect any faults in the user requirements.
5. Adaptable code generation from the SDL models
- Product code is now generated, either automatically, by means of a CASE tool, or by means of a “team member”. Because requirements differ between projects, the code generation should be adaptable and should be adjusted to suit the specific project’s needs. Automatically

generating code has the added advantage that it eliminates the need for regression testing, because all accepted validation scenarios are added to the synthesized model.

- **Automatic recovery of SDL models from legacy software**

This step describes how the above systems development methodology can be used to integrate CASE-produced components and systems into older “legacy based software”

### **Techniques**

The systems development methodology makes extensive use of formal, ITU standardised, specification languages for modeling the system, and CASE tools for reengineering existing legacy systems. Some of the formal specification languages are the Specification and Description Language (SDL) which is used to model system specifications. User requirements scenarios are captured using Message Sequence Charts (MSC). Tree and Tabular Combined Notation (TTCN) is used as a test description language, while the data is described using Abstract Syntax Notation (ASN.1). Because of these formal languages, the quality of the product is also improved (Mansurov, 2000).

### **Characteristics**

Mansurov (2000) believes that formal methods tend not to focus enough on the early stages of system development. He compensates for this “barrier to systems development methodology adoption” by implementing an “upfront increase” of both time and people. This initial increase of “approximately 17%” at the requirements phase, to “produce and validate a formal specification (e.g. in SDL)” and then to generate proper test suits at the requirements and design phases can result in 30-50% saving at the product development and deployment (Mansurov, 2000).

The three advantages of this systems development methodology is its claim to save between 30% and 50% of the project time and because of its use of formal methods for the early and later stages the quality of the product is increased, as well as the adoption level of the systems development methodology.

### **2.3.4 Summary**

Three systems development methodologies that are being used to develop telecommunication systems today have been discussed. These three systems development methodologies differ quite a lot from one another (this will be discussed in the next section), but they also differ from their more traditional counterparts. The MODA-TEL and Mansurov systems development methodologies can



be considered to be object-oriented, while Mobile-D is at least partly object-oriented. Mansurov differs from the more traditional object-oriented systems development methodologies (discussed in section 2.2.3.3) in that it does not make use of UML, even though it makes extensive use of specification languages. MODA-TEL and Mansurov both seem to have an extended focus compared to the traditional systems development methodologies. This can be seen in MODA-TEL where the focus is not only on developing a system, but actually on establishing a set of company wide “tools” that can be reused in subsequent projects. The Mansurov systems development methodology also displays this in its description of a method to recover SDL models from old legacy software. This systems development methodology also makes extensive use of automatic code and model generation. Mobile-D fits in well with the more traditional systems development methodologies, covering much of the same elements as the object-oriented systems development methodologies and also utilizing the techniques and development style of the rapid systems development methodologies. One of the aspects of Mobile-D is that it limits any modeling to sketches and short document. This is characteristic of the rapid development approach on which Mobile-D is based.

## **2.4 Comparison of Systems Development Methodologies**

These three systems development methodologies created specifically for the telecommunication industry will be compared. The aim of this comparison will be to determine the similarities and differences between the systems development methodologies, as well as to understand whether they are actually used. The framework for comparing system development methodologies, developed by Avison and Fitzgerald (2002) will be used.

### **2.4.1 Framework for Comparing Systems Development Methodologies**

The creators of this framework (Avison & Fitzgerald, 2002) give seven elements that are used to compare different systems development methodologies.

These seven elements are:

1. **Philosophy**

Looking at the philosophical assumption of any systems development methodology is “the most important aspect of comparing systems development methodologies” (Avison & Fitzgerald, 2002, p. 555).

The philosophy is the set of principles on which the systems development methodology is based. This can further be divided into four factors.

- **Paradigm**  
Does the systems development methodology fall into the systems paradigm or the science paradigm?  
The systems paradigm can be identified by the fact that the systems development methodology does not “break down the system into its constituent parts for the purpose of understanding the problems” (Avison & Fitzgerald, 2002, p. 563).
- **Objectives**  
What are the stated objectives of the systems development methodology?
- **Domain**  
Which “domain of situations” does the systems development methodology address?  
Does the systems development methodology assume that there is a specific problem that needs to be addressed or does the systems development methodology focus on identifying the systems required by the organisation?
- **Target**  
Does the systems development methodology target particular types of problems, environments or type or size organisations?

## 2. Model

The model refers to the way the systems development methodology represents important elements of the information system. The model fulfills several functions. “First, it is a means of communication; second, it is a way of capturing the essence of a problem or a design, such that it can be translated to another form without loss of detail; and third, the model is a representation which provides insight into the problem or area of concern” (Avison & Fitzgerald, 2002, p. 564).

## 3. Techniques and tools

These are any specific techniques and tools used in the systems development methodology during the development process.

#### 4. Scope

This is the phases through which the systems development methodology goes during the development process. Avison and Fitzgerald (2002) define nine stages or phases. These are:

- Strategy
- Feasibility
- Analysis
- Logical design
- Physical design
- Programming
- Testing
- Implementation
- Evaluation
- Maintenance.

These nine phases are based on the standard systems development life cycle. This presents problem because some systems development methodologies do not follow a life cycle, instead they may use an iterative model, an evolutionary or even a spiral model.

#### Outputs

These are the deliverables created at each stage of the development process.

#### 5. Practice

Practice is a measure of whether the systems development methodology is commercial or academic. How much is the systems development methodology used and by what type of user? Whether a professional analyst is required, or can the systems development methodology be used by the users themselves?

#### 6. Product

The product is what the client gets when a commercial systems development methodology is purchased.

Now that the framework has been discussed, this framework will be used to compare the three telecommunications systems development methodologies with one another.

#### 2.4.2A Comparison of Telecommunication Systems Development Methodologies

	<b>MODA-TEL</b>	<b>Mobile-D</b>	<b>Mansurov</b>
<b>Philosophy</b>			
Paradigm	It is a MDA based systems development methodology. Although the steps involved in the actual development of the system are not covered, it is known that one of the characteristics of a MDA based approach is that it is Object-Oriented and thus also falls under the science paradigm.	Mobile-D belongs to the science paradigm, seeing as it breaks up the system into parts that are then iteratively developed.	It is based on an Object-Oriented systems development methodology. According to Avison and Fitzgerald (2002), OOA falls under the science paradigm.
Objectives	The main output of the systems development methodology is to create not only the required system, but a knowledge base that can be reused in the development of other projects	This systems development methodology's focus is solely on the development of a computerized system.	This systems development methodology focuses purely on the development of a computerized system.

Domain	<p>The step described in this systems development methodology aim to setup a knowledge base which can be reused for other projects. Hence the systems development methodology does not solve a specific problem, but tries to create a base from which many different problems can be solved.</p>	<p>All the phases defined in this systems development methodology are aimed towards solving a specific problem</p>	<p>In the user requirements phase of this systems development methodology, the use-case technique is used to determine what the system is supposed to do. Use-cases determine what the interactions between different actors and between actors and the system are when a specific goal needs to be achieved. The aim of this systems development methodology is thus to solve a specific problem.</p>
Target	<p>Though the systems development methodology has been developed for distributed systems in the telecom domain, it claims to be general enough to be applied to other domains as well.</p>	<p>The Mobile-D systems development methodology is an agile approach geared specifically towards the development of mobile applications.</p>	<p>This systems development methodology has been specifically designed to “improve software engineering in the telecommunication industry”.</p>

<b>Model</b>	<p>The systems development methodology allows the developer to choose the model that should be used during the development process. This is done in the preliminary and detailed preparation phases. Even though a specific model is not given, UML is a preferred choice. Gavras <i>et al.</i> (2004) termed it Model Driven Engineering.</p>	<p>The model used during development is dependent on the architecture chosen during the “Initialize” phase. Generally the modeling is limited to sketches and short documents as needed during development. If models and documentation are needed, they are created afterwards.</p>	<p>The use of modeling is an extremely important factor in this systems development methodology. The Specification and Description Language (SDL) is used throughout the systems development methodology. It is used in modeling the user requirements together with UML Sequence Diagrams and/or Message Sequence Charts, through to the designing and code generation phases. The systems development methodology is very much model-driven. Each phase is modeled and the models are formalized before continuing.</p>
--------------	--	--	---

<b>Techniques and tools</b>	<p>Once again the systems development methodology allows the user to choose the tools they want to use. The phase in which this is done is the infrastructure setup phase. Tools are used in the systems development methodology to define models and Meta models, generate code from model information and define constraints and rules to verify model compliance. Because of the object-oriented nature of the systems development methodology, UML is given as a recommended technique.</p>	<p>During development agile techniques like test-driven development, pair programming and continuous integration are used. Other techniques include project retrospective techniques during post-iteration workshops. During the release step it is recommended that a version control tool be used.</p>	<p>A dominant technique used in the systems development methodology is to acquire user requirements by means of use-cases. Tools are also used in the transformation of user requirements to a System Description Language (SDL) document and also to generate production code, although using of tools here is not necessary.</p>
<b>Scope</b>	<p>The actual scope of the systems development methodology is dependent on the development process chosen by the developer. Most of the phases</p>	<ul style="list-style-type: none"> <li>• Analysis</li> <li>• Design</li> <li>• Programming</li> <li>• Testing</li> <li>• Implementation</li> <li>• Evaluation</li> </ul>	<p>In the systems development methodology, six phases are identified.</p> <ul style="list-style-type: none"> <li>• Analysis</li> <li>• Logical design</li> <li>• Physical design</li> </ul>

	<p>defined in the systems development methodology occur prior to the actual development of the project, but are continuously redefined during the execution phase. The execution phase is where the actual development methodology is used. In the discussion of the systems development methodology an object-oriented methodology is used and defines the following phases within:</p> <ul style="list-style-type: none"> <li>● Requirements</li> <li>● Analysis and design</li> <li>● Implementation</li> <li>● Test and integration</li> <li>● Deployment</li> <li>● Maintenance</li> </ul>		<ul style="list-style-type: none"> <li>● Programming (Code generation)</li> <li>● Automatic code synthesis</li> <li>● Testing</li> <li>● Maintenance</li> <li>● The maintenance phase is not only for newly developed systems, but the systems development methodology describes an entire process for documenting and maintaining legacy systems.</li> </ul>
--	---	--	---



<b>Outputs</b>	The main output of the systems development methodology is to create not only the required system, but a knowledge base that can be reused to develop other projects	A part of the product is completed and implemented after each iteration. Analysts can determine whether they are proceeding correctly by performing post-iteration workshops	The outputs of each stage of the systems development methodology are very specific. From user requirements, to validation, to design and code generation, all have a certain output that needs to be produced
<b>Practice</b>	This systems development methodology has academic roots, but a case study is given of MODA-TEL being used at France-Telecom	The systems development methodology was developed in practice with the help of three mobile software companies	Academic background. It is not known if the systems development methodology is used by any company

**Table 2.1: Comparison of mobile telecommunication system development methodologies**

### 2.4.3 Conclusion

After comparing the three telecom systems development methodologies discussed in section 2.3, using the comparison framework by Avison and Fitzgerald (2002), the following conclusions can be made:

- **Similarities**

All three systems development methodologies first break down the system to be developed into more manageable pieces. Each modular piece is then developed separately and is later combined to form the complete product. This style of modular development greatly reduces the overall complexity of the system.

All the systems development methodologies were designed specifically for the telecommunication domain and are used in the development of a computerized system.

There are many similarities between the MODA-TEL and Mansurov systems development methodologies. One similarity is that the need to model the system is a very important element of both MODA-TEL and Mansurov. They also use techniques that are typical of object-oriented systems development methodologies like making use of UML, in both MODA-TEL and Mansurov, and use-case techniques as in Mansurov.

When looking at the scope of the different systems development methodologies, it is noted that the phases defined for Mobile-D and Mansurov are much the same, each going from analysis to the maintenance phase.

All three the systems development methodologies describe specific outputs during the development process. MODA-TEL aims at creating the end product as well as a knowledge base that can be reused to develop other projects. Mobile-D's main outputs are the product parts that are released after each iteration. Mansurov also clearly defines the outputs after each stage of development. Mostly these outputs consist of specification documents.

- **Differences**

While all the systems development methodologies are used to develop a computerized system, the MODA-TEL systems development methodology also aims to build a knowledge base that can be reused in the development of other projects. This means that the MODA-TEL systems development methodology is not only focused on solving a specific problem, as is the case with the other two, but to improve business by building a reusable knowledge base.

The systems development methodologies discussed were all developed for the telecommunication domain, but while Mobile-D and the systems development methodology created by Mansurov did not mention anything about its use outside the telecoms domain, MODA-TEL states that it is general enough to be used in other domains as well.

While the need to model the system is an important element of both the MODA-TEL and Mansurov systems development methodologies, Mobile-D is less concerned, limiting any modeling to sketches and short document. This is characteristic of the rapid development approach on which Mobile-D is based. The same is true of the techniques used during development. Typical object-oriented

techniques are used in MODA-TEL and Mansurov, while Mobile-D uses typical XP style techniques like test driven development, pair programming and continuous integration.

The scope of Mobile-D and Mansurov is much the same; MODA-TEL differs though because the scope is dependent on the chosen system development process.

Although the three systems development methodologies have been developed with the same goal in mind, namely for developing telecommunications systems, there are many differences between them. The biggest difference is the fact that two of the systems development methodologies follow an object-oriented approach, while the other is a mix between object-orientated and rapid development systems development methodologies.

## **2.5 Mobile Telecommunication Environment**

In order to determine what the needs and challenges are when developing mobile telecommunication software, a brief study was done of the mobile telecommunications environment. This discussion will help in answering the research question to “identify a list of necessities a system development methodology should contain, when used to develop mobile telecommunications software”. The results are discussed below; first the environmental elements will be discussed, followed by some of the specific needs in mobile telecommunication system development. It will then be determined if the methodologies discussed above adhere to these elements and to what degree. This will show if the systems development methodologies are able to address these environment specific elements.

### **2.5.1 Software Development Needs**

Big telecommunication companies like Bell Canada and Northern Telecom outsource most of their product development needs to other companies (Abbas, Kazmierczak, & Dart, 1998). To determine the out-source company's product development and support capabilities, they developed the Trillium development model (Coallier, Graydon & Ficcici, 1995).

This model discusses most of the processes, procedures and activities that should be used to improve the telecommunications product development process or life cycle. It relies heavily on industry standards like ISO, Bellcore, Malcolm Baldrige, IEEE and IEC. According to Vallecillo (2001), the use of standards is an important factor in telecommunication system development.

By following this model developers can ensure that the cost of development and maintenance is minimized and it also helps to decrease the development time, meet objectives and ensures user satisfaction. According to Coallier, Graydon and Ficcici (1995), a systems development methodology used for developing mobile telecommunications systems should :

- Have a definite and well-documented analysis and design phase
- Verify specified items by means of prototyping
- Use CASE tools to automatically generate code, prototypes and specifications
- Use formal methods when critical software components are being developed
- Use formal document notation to document requirements, the design of the product, code generation, test cases and maintenance plan
- Verify the adequacy and efficiency goals of the product by means of strenuous testing and review sessions
- Facilitate the reuse of source code and/or components of the development process. The reuse of source code can be aided by keeping a component repository. Reuse is also facilitated by means of product line reuse strategies and CASE tool support.

### 2.5.2 Telecommunications Environment

There are quite a few elements that characterize the mobile telecommunications environment. Some of these elements will be discussed below, with the key elements displayed in bold.

Telecommunications systems are **distributed** by nature (Gervais & Ruffel, 1997) with software components spread across the network. This distribution is caused by the physical distribution of the users as well as service providers (Patel, 2002). Telecom systems are made up of many components in different locations running separately and in parallel. According to Patel (2002), these additional problems of “physical component distribution and dynamic concurrent behaviour” cannot be handled by existing software engineering tools.

When designing telecom software, special attention should be paid to **scalability**. This is due to the fact that telecommunication networks are very dynamic and tend to grow/change at a rapid pace (Patel, 2002; Golubic & Marrusic, 1999; Mehra, 2005; Lal, Pitt, & Beloucif, 2001; Holler, 2006). This **rapid growth** means that there is a constant introduction of new services and products and an increase in the need for cross system integration. Because of this any developed software should be as **flexible and adaptable** as possible (Mehra, 2005).

A key requirement of any telecom system is **fault-tolerance**. Unfortunately, dealing with this issue causes increased complexity (Patel, 2002).

Telecommunications software is known for being **very expensive** (Patel, 2002). This is caused by the complexity of the systems, but also because few of the objects created during previous projects are **reused** (Mansurov, 2000). This is most noticeable in large companies where different teams could be working on solving exactly the same problem. A **proper communication medium** is clearly needed to assist in knowledge transfer (Patel, 2002).

Due to the **real time nature** of telecommunication systems, handling **fault-tolerance** and **safety** in telecommunication systems often becomes complicated and this increases the **complexity** of the whole system (Patel, 2002). Several factors can be identified to be the cause of this complexity. Firstly the constraints imposed on the **time** and **available space** of the development are big contributing factors, secondly, the highly complex time-critical elements that are present in telecom systems, thirdly, the automated failure detection and handling elements (Patel, 2002). The increasingly ambitious and complex requirements imposed on telecom systems (Kalayanasundaram *et al.*, 1998) are a fourth reason for its inherent complexity.

The issue of **security** is also very important, especially when considering e-business applications (Patel, 2002).

Because these elements are important factors in the mobile telecommunications industry, they should be addressed during the development of mobile telecommunications systems. From the above discussion the following elements can be added to the list of software development needs that should be addressed by the systems development methodology used.

A systems development methodology used for developing mobile telecommunications systems should:

- Take into account the distributed nature of mobile telecommunications systems and facilitate the building of such systems
- Focus on producing scalable products to handle the growth and change inherent in the industry efficiently
- Have products which are flexible and adaptable enough to handle the introduction of new services and networks.

- Help to reduce the cost of the project
- Define a proper communication medium that can be used to transfer knowledge within the company
- Be able to handle and even reduce the complexity of the product being developed
- Reduce the time needed to development the system.

### 2.5.3 Systems Development Methodology Discussion

In this section how the three telecommunications systems development methodologies adhere to the element discussed in the above two sections will be viewed. This is done in table 2.2, displaying the criteria down the left column with the three systems development methodologies' adherence to the criteria discussed in the following three columns. The cells of the table contain the verdict, as well as the discussion of the intersecting criteria and systems development methodology. The discussion in the cells sees whether the criteria is always adhered to (Yes), is never adhered to or not mentioned (No), or is sometimes adhered to or briefly mentioned (Maybe).

	MODA-TEL	Mobile-D	Mansurov
<b>Software Development Needs</b>			
Well-documented analysis and design phase	<b>Yes.</b> The systems development methodology specifically states that during the requirement and analysis phases of the system development process, all models should be well defined and documented	<b>Yes.</b> There are definite analyses and design phases for the entire project as well as for each "productionize" iteration. These are documented using requirements documentation and user interface illustrations	<b>Yes.</b> Analysis and design is a focus area of this systems development methodology. The use of formal specification languages to document the user requirements, analysis and design data are mandatory

Prototyping	<b>No.</b> Although prototyping can be incorporated in the chosen systems development process, no mention is made of it in the systems development methodology itself	<b>No.</b> Prototyping is not used in this systems development methodology	<b>Maybe.</b> Automatic generation of prototypes from formalized models are mentioned, but not required
Automatically generate code, prototypes and specifications	<b>Yes.</b> During the preliminary preparation phase, the developer is required to define a method for transforming abstract models into concrete models. It is not required that models be generated automatically. During the project execution phase though, the systems development methodology does mention the use of model transformation and code generators	<b>No.</b> All specifications are developed manually, while code is developed by means of pair programming	<b>Yes.</b> All specifications are formalized. SDL models are automatically generated from MSC models. After this the SAM model is generated from SDL which in turn is used to generate the product code automatically

Formal document notation	<b>Yes.</b> During phase 2 of the systems development methodology, a modeling language is defined for representing data, business processes, user requirements etc	<b>No.</b> No formal document notation is defined for this systems development methodology	<b>Yes.</b> Formalised documents and specification languages are a main focus area for this systems development methodology
Strenuous testing and review sessions	<b>Yes.</b> Once again, this depends on which systems development process is chosen during the project management phase. The systems development methodology, however, does suggest that testing should be done and well modeled, utilising industry standards	<b>Yes.</b> The whole development process is test driven. Before releasing code it is tested, and there is also a system test and fix phase that iteratively tests the system and fixes any problems	<b>Maybe.</b> Throughout the project the specifications are validated and are used to generate test cases automatically. Even though test cases are developed, no mention is made of specific testing or review sessions. The need for testing is somewhat eliminated when the product code is generated automatically
Reuse	<b>Yes.</b> Although MODA-TEL does not specifically talk about reusing code, one of the systems	<b>No.</b> There is no mention made of reuse in the systems development methodology	<b>No.</b> Some parts of specification documents can be reused, but other than that no mention is



	development methodology's main focus points is on defining and reusing components of the development process		made of reuse in the systems development methodology
Distribution	<b>Yes.</b> Specifically designed for developing distributed systems. This can be deduced by the fact that MDA is based on platform technologies such as CORBA which was specifically created for distributed systems	<b>Yes.</b> This systems development methodology has been designed for mobile systems, with its inherent distribution	<b>Maybe.</b> The systems development methodology does use models and languages that have been standardised for the telecommunications industry, but it does not specifically take distribution into account
Scalable product	<b>Yes.</b> Technology independent modeling quickly and easily implements any new system elements or technologies	<b>Yes.</b> During the explore phase plan is specifically developed for handling system expansion and growth	<b>Yes.</b> The systems development methodology also discusses how existing systems can be modeled for redevelopment and enhancements

Flexible and Adaptable product	<b>Yes.</b> Meta-data repositories are created which assist in dynamically discovering new services and making them available to the clients	<b>Yes.</b> During the explore phase the plan is specifically developed for handling system expansion and growth	<b>Yes.</b> Adaptable code generation procedures help to keep the product as well as the development process flexible and adaptable
Reduce cost of project	<b>Maybe.</b> The use of MDA model is credited with reducing overall development cost	<b>Yes.</b> Techniques used such as refactoring can reduce development cost	<b>Yes.</b> The use of formal CASE tools as well as the upfront increase in project time and people reduces the total cost of the project
Communication medium	<b>Yes.</b> UML is used to communicate the system throughout the company	<b>No.</b> No formal documentation or modeling language is used	<b>Yes.</b> Everything is formally documented and modeled and these models are used to communicate the requirements throughout the company
Reduce complexity of product	<b>Yes.</b> The project follows an established model based approach and first breaks up the development into modular pieces. All this contributes to	<b>Yes.</b> The system is first broken up into modular pieces. Using the refactoring technique during development also decreases the	<b>Yes.</b> Everything is formally documented and modeled. This reduces the overall complexity of the development. Also, the system is

	eliminating complexity	complexity	divided into modular pieces
Reduce development time	<b>Maybe.</b> The use of MDA model is credited with reducing the systems development time	<b>Yes.</b> The focus of the systems development methodology is on speed of development. Another contributing factor is that it uses very few documents	<b>Yes.</b> The use of formal CASE tools as well as the upfront increase in project time and people can speed up the time-to-market by 20%-30%. But on the other hand it could be said that having to generate all the models is time consuming. This is negated by the fact that the models are automatically generated

**Table 2.2: Mobile telecommunication system development methodologies adherence to list of software development necessities**

#### **2.5.4 Summary**

The three telecommunications systems development methodologies were compared to a list of software development needs and environmental criteria. This list could hold true for all computerized systems development, telecommunication or otherwise, but as stated in Jantsch *et al.* (1998), "the difference between the systems in different domains is usually not the absence or presence of characteristics but the degree of importance of different characteristics". From this it can be seen that for the most part, the systems development methodologies adhere to the criteria identified for developing telecommunication systems.

## **2.6 Conclusion**

In this chapter a systems development methodology was defined and a literature study on the systems development methodologies that are available when developing a computerized system was done. Three systems development methodologies that were developed specifically for the telecommunications domain were discussed. These three systems development methodologies were then compared using the framework suggested by Avison and Fitzgerald (2002) in order to identify any differences or similarities between them. The comparison also gave more insight into whether and how the systems development methodologies are used in systems development. The last part of the chapter looked at different elements of the telecommunication environment to identify some of the needs and challenges faced when developing mobile telecommunication software. The three telecommunication systems development methodologies were then checked to see how they adhered to the elements of the above list. In the next chapter the research method used in this study will be given as well as a discussion on other available research methods.

## CHAPTER 3

### 3. RESEARCH DESIGN

#### 3.1 Introduction

When doing research on the subject of systems development methodologies, there are a few different research approaches that can be followed. The type of approach used depends on different elements of the study, such as the research question and also the research method that is used. The approach that was followed in this study was the qualitative research approach.

In this chapter, the different aspects of research design will be discussed, beginning with the research paradigm, followed by a discussion of the different research approaches. After this, some research methods are given, as well as data collection, data coding and finally data analysis methods.

#### 3.2 Research Paradigm

When doing research, the type of research done depends on the research paradigm. A research paradigm, according to Myers (2003), is built on underlying assumptions about knowledge and how that knowledge can be obtained. The knowledge relates to the nature of the problem at hand and the assumptions made are used to set the direction for the research. There are three research epistemologies that can be followed when doing research (Chua, 1986), the positivist, critical and interpretive.

**Positivist research** assumes an objective reality with measurable properties, and that these properties are totally independent of the researcher. Positivist studies are generally made to test current theories.

Rather than understanding and explaining a theory, **critical research** aims to give critique on and change the society in which the study is done, while taking into account any social, political and cultural domination.

**Interpretive research** tries to understand the subject being researched by understanding the meanings that people assign to them, focusing on the complexity of the research domain and the people within it.

In this study interpretive research was conducted. The assumptions made about the study that led to the use of interpretive research were that the author did not set out to prove a theory or change the target domain, but rather to understand the telecommunication environment with all its inherent complexities, and in so doing, find out whether systems development methodologies are used during system development, and if so, how and why are they used.

### **3.3 Research Approach**

There are two main research approaches, quantitative research and qualitative research. The difference between these two is that quantitative data is represented by numbers, while qualitative data is represented by pictures and words (Gilgun, 1992). Besides the difference of words and numbers, there are a few other differences between the two research approaches. Six differences between qualitative and quantitative research methods, as found in Sharp and Frechtling (1997) will be discussed.

- Value of data

The information gathered through quantitative research is more representative and broadly generalisable. A drawback is that when focusing on a specific issue within the study, the reasons given by the survey are often few, and not very useful (Sharp & Frechtling, 1997).

For example: A study shows that, because of a science enrichment programme, pupils will be more inclined to do their masters degree in science. But the study cannot give specific reasons why the results differ according to gender.

Qualitative research overcomes this challenge by limiting the survey to specific focus groups. This has the drawback that the findings do not necessarily apply outside the focus group.

- Scientific rigor

Scientific rigor can be explained by how accurate the data is believed to be.

When data is collected using quantitative methods, standardised statistical techniques are used. It is believed that this leads to more objective and accurate information which is usually highly detailed and structured (Neville, 2005).

It has also become clear that all data collection is in some way affected by the beliefs and perceptions of the person collecting the data (Sharp & Frechtling, 1997).

The issue still remains that qualitative results are often considered less conclusive than quantitative results (Seaman, 1999).

- Philosophical distinction

Researchers differ about the nature of knowledge and what is the best way in which this knowledge can be acquired (Sharp & Frechtling, 1997).

Qualitative researchers try to understand the context in which certain events occur as opposed to quantitative researchers who “seek to control the context” (Sharp & Frechtling, 1997).

- Credibility of findings

This refers to the feeling of the research audience towards the type of results produced by the research.

The numbers and statistical results of quantitative research are often favoured by decision makers. On the other hand, others might prefer the “richer data obtained through qualitative research” (Sharp & Frechtling, 1997). Another issue with qualitative data is that can be challenged easily (Neville, 2005).

- Staff skills

Quantitative research is harder to design initially (Neville, 2005) and good skills are required in qualitative research methods like observations, interviews and focus groups, while quantitative methods are easy to master (Sharp & Frechtling, 1997).

- Time constraints

Because of the overlapping processes of data collection and analysis, qualitative methods are more time consuming than quantitative methods (Sharp & Frechtling, 1997).

Using qualitative methods is recommended for exploratory research where there is a lack of literature (Creswell, 1997; Neuman, 2003), as is the case in the current study. For this reason the qualitative research approach was used during this study. Another reason is because human behaviour, together with technical issues and human-computer interaction, plays an important role in software development (Seaman, 1999) it is import that the research method used in this study takes this into account. Qualitative research methods do this because they are known for their use when studying the complexities of human behaviour (Seaman, 1999).

## **3.4 Research Method**

When doing qualitative research, there are many different research methods that can be followed. Three research methods will be discussed below: ethnography, grounded theory and case studies. In this study the case study research method was used. Therefore, case studies will be discussed in more detail.

### **3.4.1 Ethnography**

When using “Ethnography” as a study method, the aim is to make a study of an entire culture. In fact, the researcher needs to incorporate himself into the environment being studied. Doing this allows the researcher to study all aspects of the culture and this is important because “properties cannot necessarily be accurately understood independently of each other” (Trochim, 2006).

With this technique, the main method of collecting data is participant observation (Hussey & Hussey, 1997).

### **3.4.2 Grounded Theory**

Grounded theory is an “inductive theory discovery that allows the researcher to develop a theoretical account of the general features of the topic while simultaneously grounding the account in empirical observations of data” (Martin & Turner, 1986, p. 141).

There are two key factors when using grounded theory, these are that the research should not start with a theory which must be proved or disproved. The second is that grounded theory is discovered by comparing the properties of different categories and finding “uniformities and diversities” (Fernández, 2004, p. 84). The way in which data is acquired is mainly by means of semi structured interviews or observations (Seaman, 1999).

### **3.4.3 Case Studies**

One of the methods used in qualitative research is the case study method. By using the case study method a researcher can gain an understanding of a complex issue or object. There are a variety of data gathering methods that can be used to collect data, a few examples being interviews, documentation review, observation and the collection of physical artifacts (Soy, 1996). Case studies can be used to explore new areas of research where little theory is available. They are also used to describe a process, or the effects of an event, or even explain a complex phenomenon (Kohn, 1997). It is an extremely powerful method in its ability to answer questions of how and why (Yin, 2003).



### **3.4.4 Qualitative research method used in this study**

It was decided to make use of the case study research method. According to a study done by Soy (1996), there are six steps to follow when using the case study method. These six steps as well as how they were implemented during the study will briefly be discussed.

#### **3.4.4.1 Case study steps**

- **Determine and define the research question**

Firstly, the researcher should establish a research focus. This is done by forming how and why type questions of the research area and determining the purpose of the study. Defining these questions also helps to determine the type of analysis that shall be used later in the study.

The focus of this study is on systems development in the mobile telecommunication industry. The main objectives are to identify the systems development methodologies used, if any, by companies in South Africa in the development of telecommunication systems. To highlight certain key aspects of mobile telecommunication software development and identify a list of necessities a systems development methodology should contain when used to develop mobile telecommunications software.

In order to gain a deeper understanding about this subject, the choice was made to use the case study method.

- **Select the cases and determine data gathering and analysis techniques**

When designing the case study research, the researcher must determine the approaches which will be used in selecting the case study, the instruments necessary as well as the appropriate data gathering methods. Good choices on these subjects will positively increase the validity of the study (Soy, 1996).

When choosing the cases for the study, the author wanted to gather information from sources that had experience in the development of mobile telecommunication software systems. Another important factor was to choose cases that had different characteristics such as size and the types of systems developed. This would increase the scope of the case studies to cover as much of the telecommunications domain as possible.

Three companies were chosen as subjects for the case study. A short description of the three companies follows. They will be discussed in more detail in Chapter 4:

**Company 1** is a big mobile telecommunications service provider and software development centre with many departments and employees who develop a wide range of different types of telecommunications systems.

**Company 2** is a medium sized company with about 30 personnel. The systems they develop are focused more on mobile entertainment services (like ringtones, logos, picture messages, etc.), incorporating SMS, MMS and web based “WAP” services (including m-commerce systems).

**Company 3** is a very small company with only 10 employees. Their main focus is on developing GSM base systems as well as hardware telemetry.

- **Prepare to collect the data**

This step entails that the researchers should make themselves comfortable with the chosen data gathering methods. Often a pilot test will be conducted to uncover any issues.

To prepare for the data collection the researcher did a study on the different methods that will be used, these being on the subject of interviews and different data analysis methods. In this study interviews were chosen as the method in which data was collected. A pilot test was not done because of the author’s previous experience in conducting interviews.

- **Collect data in the field**

The subjects of the case study were chosen, being three companies from different areas of the telecommunication industry, and interviews were conducted at each company. The interviews were conducted with project managers and developers, and generally took between one and two hours. Interviews were recorded using a laptop and microphone and were immediately transcribed. This data was then stored in such a way that lines of inquiry and patterns could be uncovered. Field notes, taken to record the researcher's feelings and hunches also contributed to the interview data and served as a base for more questions. It also served to document the work progress.

- **Evaluate and analyze the data**

The data should now be analyzed with reference to the original research questions. The data is sorted in many different ways to expose or create new insights (Soy, 1996).

Techniques for analyzing data include creating matrices of categories, creating flow charts and tabulating frequency of events. Another technique is the cross case search for patterns.

After the coding of the interview data and interviewer notes using ATLAS.ti, the cross case analysis method was used to analyze the data.

This technique will be discussed in section 3.7.2 “Cross Case Analysis”.

- **Prepare the report**

These complex issues should now be converted into something which is understandable.

Special attention should be given to display sufficient evidence for the findings while also pointing out conflicting propositions. The boundaries of the case should be defined and it should also be shown that all avenues have been explored.

There are several ways to compose the report, the researcher can handle each case separately, recount the cases chronologically or represent the case studies as stories.

To report the findings of the case study the author decided to focus on the main cases and handle each of them separately.

#### **3.4.4.2 Arguments against case studies**

Some of the arguments made by critics of the case study method are that reliable findings cannot be established by studying a small number of cases (Soy, 1996). The reason for this is that the researcher cannot be sure that the case studied can be generalized and that it has only limited representativeness (Bryman, 2004). This issue of generalisability and representativeness is also mentioned in Ruegg (2006) and Hancock (1998) as being a problem inherent in case study research. Another issue is the feeling of some critics that findings are often biased. The inability to stay removed from the subject under observation is considered to be a very limiting factor in case study research and it decreases the validity of the results (Yin, 2003). Lastly is the belief that case studies should only be used as an exploratory tool (Soy, 1996).

#### **3.4.4.3 Arguments for case studies**

The case study method is thought by Voss, Tsikriktsis and Frohlich (2002) to be one of the most powerful research methods available, particularly when exploring a new research area or developing a new theory. This is confirmed by Darke *et al.* (1998) and Eisenhardt (1989), who suggest that the case study method be used when researching newer less well-developed research areas. Walsham (1993) also states that the case study method is the most appropriate method to use when conducting interpretive research. Another argument for using case studies is given by Burns (2000), who states that case studies generate rich data that give intricate detail of the subject being researched.

After considering the above mentioned arguments, the case study method was chosen to be used in this study. The reason for this choice is the overwhelming support for case studies when doing exploratory research. To ensure that the case study was performed correctly, the author adhered to the principles for interpretive field research, as discussed in Klein and Myers (1999).

#### **3.4.4.4 Principles for interpretive field research**

In Klein and Myers (1999) the authors discuss seven principles to adhere to when doing interpretive case study research on Information Systems. Adhering to these principles helps the researcher to better understand the case which in turn leads to a better understanding of the field study (Klein & Myers, 1999). It also serves as a means to evaluate interpretive research. The use of the principles is not mandatory though and it is up to the researcher to decide which principles are adhered to, and to what degree. To ensure the validity of the interpretive case study, the author strived to adhere to these principles. The seven principles are (Klein & Myers, 1999):

- 1) The fundamental principle of the hermeneutic circle  
This is the one fundamental principals of interpretive research. It is the idea that to understand the research subject, one should constantly be looking back and forth between the whole and its parts.
- 2) Contextualization  
To better understand the information being gathered, it is necessary to understand the social and historical context in which the study takes place. Identifying the context identifies how the situation that is being studied emerged.

3) Interaction between the researchers and the subject

The data that is gathered during the case study is a product of the social interaction between the researcher and the participant. It should be noted that the information given by participants is based on how that information is interpreted by the participant.

4) Abstraction and generalization

The validity of the cases depends on the logic within the descriptions of the cases and the final conclusions. One also needs to relate the information gathered by means of the case study to “theoretical, general concepts that describe the nature of human understanding and social action” (Myers & Klein, 1999).

5) Dialogical Reasoning

Because the findings gained from a case study are often bias, any preconceptions that the researcher might have should be identified and, if not supported by the findings of the study, abandoned.

6) Multiple interpretations

The researcher should be aware that a single subject could be interpreted differently by different respondents. It will also be prudent to document the reason for this difference in interpretation.

7) Suspicion

The researcher should look at the information gathered in a critical manner and not take the meaning of the responses to be the only truth. Any data should first be looked into more deeply.

### **3.5 Data Collection Method**

There are several different ways of obtaining qualitative information, some examples are individual focus groups, observation and interviews. In this section these three methods will be discussed followed by a description of the method used in this study.

### **3.5.1 Participant Observation**

During participant observation, the researcher has to be on site to gather the required information. The information gathered during participant observation is valuable when describing settings, activities, people and the observations from the participants' perspective (Hoepfl, 1997).

Participant observation has an advantage over interviews in that it leads to a deeper understanding of the context in which events occur. Both verbal and non verbal cues are observed and recorded. Unfortunately only a small part of the software development process can actually be observed (Seaman, 1999).

Researcher participation is a big issue during observations, because the presence of an observer will likely introduce distortion (Hoepfl, 1997).

### **3.5.2 Focus Groups**

Focus groups are interviews with small groups of people, typically about 6 to 10 (Hancock, 1998). This method uses the knowledge and interaction of the group to gain insight into a subject that would otherwise not have been gained without group interaction (Mahoney, 1997). Focus groups can only be used when the subject matter is not so sensitive that the respondents will withhold information or alter responses because of the group situation (Mahoney, 1997).

### **3.5.3 Interviews**

Interviews are a more personal approach than focus groups, conducted with individual persons. Doing interviews is a good way of gaining data when the subject matter is very complex and there is a greater volume of issues that need to be covered.

### **3.5.4 Data collection method used in this study**

During this study interviews were used as the method for data collection. For this reason the author did an extensive study on interviews and interview methods. This study is discussed below.

#### **3.5.4.1 Why Interviews**

The reason why the use of interviews was chosen as the method for data collection is because according to Walsham (1995), interviews are the best method for collecting data when doing interpretive case studies. Rubin and Rubin (1995) state that if the aim of the research is to find out what others feel and think about their world, which is the case in this study, interviews are the most effective data collection method.

Some further advantages of face to face interviewing, according to Thomas and Brubaker (2000) are:

- The fact that the interviewer takes time to conduct a personal interview, instead of just sending around a questionnaire, tells the respondent that the interviewer values the respondent's opinion
- This in turn can lead to better data being given
- The interviewer is there to clarify anything the respondent might find confusing
- Interviews give the opportunity for the respondent to stress certain topics, or digress from the central topic of conversation
- Interviews can provide more understanding of the respondent's motives, reasoning and emotional reaction.

#### **3.5.4.2 The Interview**

Interviews are special forms of conversation that provide a way of gathering data by talking to people about a certain topic.

There are different types of interviews, each with its own advantages.

- Loose approach (Thomas & Brubaker, 2000): In this type of interview questions are posed in a very general form and the interviewee is then given unrestricted freedom to answer the question. This approach is mostly used to find out how different people interpret certain questions. This type of interview is also known as an unstructured interview (Seaman, 1999).
- Tight approach (Thomas & Brubaker, 2000): This approach gives the interviewee the opportunity to choose a response from a list of possible answers. This style is very much like a questionnaire. Sometimes the interviewer might ask the respondent to give a reason for their response. Unfortunately this style constrains the interviewee too much, and is not very useful for qualitative interviewing. One advantage is that the results of the study can be compiled more easily. The tight approach is also known as a structured interview (Seaman, 1999).
- Converging approach (Thomas & Brubaker, 2000): Also known as a semi structured interview (Seaman, 1999), combines the loose and tight approaches. The interviewer starts by asking general, open ended questions. Depending on the answers the respondent gives, the interviewer could then ask one or two limited choice questions.

- Some of the advantages and disadvantages of using the converging approach, according to Mahoney (1997) are:

#### **Advantages**

- The data gathered is usually very detailed and could lead to new insights
- The interviewer gets the opportunity to explore certain topics in more depth if required which increases the likelihood of useful responses
- The interviewer is allowed an amount flexibility to adapt the interview to particular circumstances.

#### **Disadvantages**

- Data gathered might be distorted because of recall error, selective perceptions or the interviewee's desire to please the interviewer
- Flexibility can result in inconsistencies across interviews
- The amount of information gathered during interviews could be very large, which could lead to difficulties in transcribing the data.

It was decided to use the converging, semi structured approach because of the advantages such as that the converging approach lends itself to an exploratory study, in particular the fact that open ended questions allow for expected and unexpected information.

The interviews were conducted as follows.

#### **3.5.4.3 Setting up**

According to Mathers, Fox, and Hunn (1998), the first thing to do is to locate a respondent. Of course one should choose a respondent who has knowledge about the topic being researched. The timing of the interview, as well as the location, should be chosen to best suit the interviewee. After choosing a respondent, a request must be sent to obtain the respondent's consent to partake in the study. "The interviewer must reassure the respondent of confidentiality or anonymity, and inform them that identities will not be revealed in the aggregated findings" (Mathers, Fox, & Hunn, 1998, p. 10).

In this study it was decided to conduct the interviews at different telecommunication companies. These companies were chosen as being representative of the current mobile telecommunications environment, taking into account the development of different software types as well as different



company sizes. Contact was made with the IT manager of each company and asked whether they would be willing to make a developer or project manager available to partake in the study. Only three of the companies contacted agreed to the interviews.

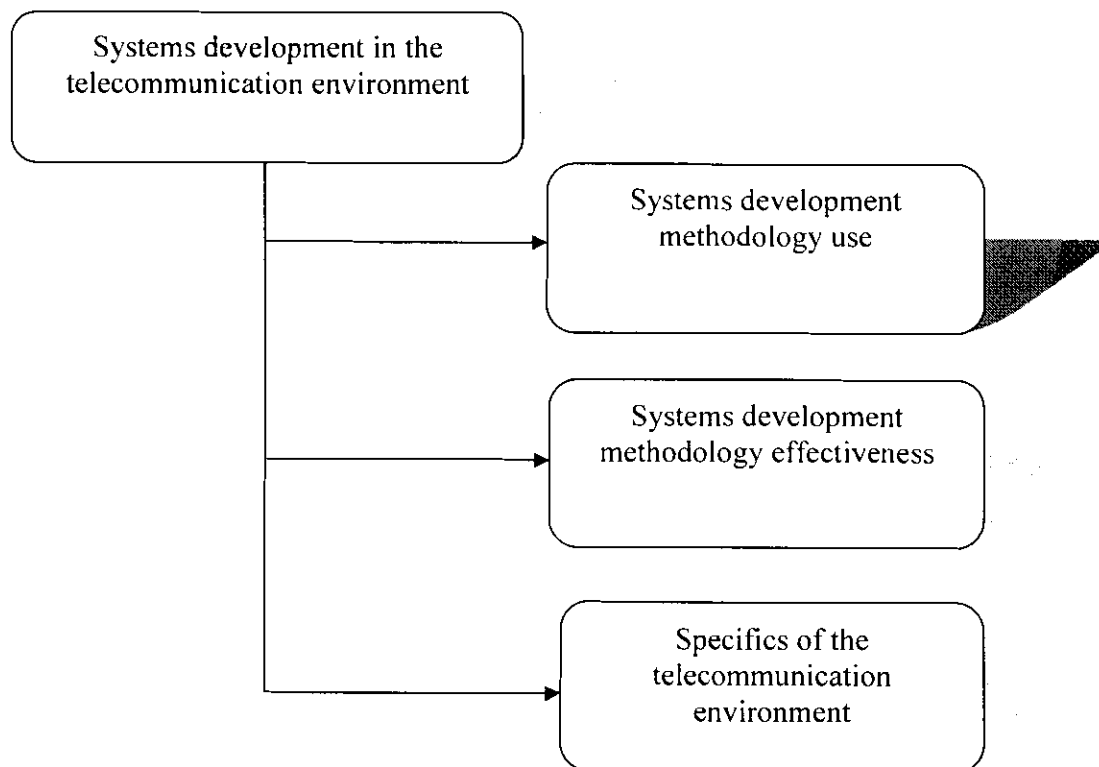
Two of the three companies interviewed specifically stated that they have signed non-disclosure agreements and would not be able to divulge any sensitive information.

After the companies at which the case studies would be done were contacted, the interview was designed. There were certain decisions that had to be made concerning what questions to ask, how they should be phrased and in what sequence to ask the questions (Mathers, Fox, & Hunn, 1998). The author also had to be very clear about depth and breadth of topics that would be included in the interview. Mathers, Fox and Hunn (1998) list 12 guidelines to remember when setting up an interview. These guidelines were utilized in setting up the interview questions for this case study.

- Questions must be answerable
- Leading questions should be avoided
- It is important that the interviewer keep in mind with what perspective the interviewee responds to questions, such as whether the respondent has first hand experience, or picked it up from a third party
- The time an interview takes should be kept to a minimum
- Avoid using words and phrases that the interviewee will not understand
- Realize that some words might have different meanings depending on the geographical area or culture
- Know that some words are very subjective and the interviewer should always ascertain what the interviewee means with words like “good” or “satisfactory”
- Some interviewees might not be able to provide answers about the full range of issues covered
- Interviewees might have a range of different perspectives
- The first question should be something the interviewee knows and can answer with ease. Personal information about the interviewee is good to start with
- Supplementary questions should be designed to prompt and probe the interviewee to extend the responses to the main questions

- Summarizing or recapping near the end of the interview signals the end of the interview. The interviewee should be invited to correct anything the interviewer might have misunderstood.

When setting up the interview questions the author looked at the main research question stated at the beginning of the study, namely how is software developed in the mobile telecommunications industry. This main research question was then divided into mini research questions, as illustrated in Figure 3.1, which focused on systems development methodology use and effectiveness as well as specifics of the telecommunication environment. Systems development methodology use focuses on what methods and procedures the company uses when developing a new system, and also whether a formal systems development methodology is used. When looking at systems development methodology effectiveness, the focus is on whether the systems development methodology is effectively used. The last mini research questions looks at the telecommunications environment and tries to determine any specific elements in the environment that influence the development process. All the questions were designed to be as open ended and probing as possible and to take into account all the guidelines discussed above.



**Fig.3.1 Research design.**

#### 3.5.4.4 Interview Questions

In this section the questions that were asked during the interviews, as well as the motivation for asking the specific questions will be given. These were the “base questions” that were asked during the interviews and when required, further questions were asked to obtain more information on certain issues. The first three questions asked were general questions, which were specifically chosen to be questions the interviewee could easily answer and to set the platform for a conversational style interview, yet also provide some background information on the company and interviewee. The remaining questions were then chosen based on the research design discussed above in section 3.5.4.3. These questions focused on gathering information about systems development methodology use, systems development methodology effectiveness and also specifics of the telecommunication environment.

##### General Questions

Questions:	Motivation:
Tell me a bit about your company? (Size, philosophy, age, etc.)	Get more information specific to the company. Some companies focus only on development, while others might focus on solutions and maintenance
What is your position in the company and what are your main tasks?	This might cause different viewpoints of system development. For instance, a developer, an analyst and a project manager could all have different knowledge about different aspects of system development
How many years experience do you have?	The level of the interviewee's experience gives some background information on the interviewee

**Table 3.1: General questions asked during interviews**

**Mini RQ 1: System Development Methodology use**

<b>Questions:</b>	<b>Motivation:</b>
What type of systems do you develop? (Size, function, etc.)	Different types of systems could lead to different methods of development
Tell me about the different phases and processes you follow during systems development	This question is to gain more understanding of the method they follow to develop a system. Depending on the interviewee's answer to this question, follow-up questions were asked to gain further information on techniques, tools and documentation used during specific phases or processes
Are these formal processes or were they developed in house?	The use of formal processes could be a sign that a formal system development methodology is being used

**Table 3.2: Questions asked during interviews on system development methodology use****Mini RQ 2: System Development Methodology effectiveness**

<b>Questions:</b>	<b>Motivation:</b>
Are you required to develop a system using these phases, processes and techniques, or do you use them of your own accord?	When the developer is required to use a specific methodology, he/she has to use it, regardless of any faults he/she might find in it, which leads to the next question
How do you feel about the effectiveness of these processes?	This answer could lead to great insight into how certain problems in telecommunication system development should be solved
How would you alter the development process to increase effectiveness?	Once again, this answer could lead to great insight into how certain problems in telecommunication system development should be solved
Could you elaborate on any specific part during systems development that is	This identifies the problems inherent in the telecommunication system development process

difficult or problematic?	and also leads to the next question
How are these problems solved?	This could identify certain processes that are being used, or could be used, as well as how effective they are at solving the proposed challenges
How successful are your development projects?	This is a good measure of the effectiveness of the development process. The interviewer had to keep in mind that there is a chance of this question not being answered truthfully
How successful are your development projects?	This is a good measure of the effectiveness of the development process. The interviewer had to keep in mind that there is a chance of this question not being answered truthfully

**Table 3.3: Questions asked during interviews on system development methodology effectiveness**

**Mini RQ 3: Specifics of the telecommunication environment**

<b>Questions:</b>	<b>Motivation:</b>
Do you have experience in the development of software other than telecommunication software?	If the interviewee's knowledge is limited to telecommunication systems, it could have an effect on the answer to the next question
In your opinion, how does telecommunication software development differ from other types of software?	This question identifies the specific needs of the telecommunication architecture
What do you feel are the effects of these differences on the development process?	This question tries to establish whether a systems development methodology used for developing a telecommunications system will differ from other types of development based on industry differences

**Table 3.4: Questions asked during interviews on specifics of the telecommunications environment**

#### **3.5.4.5 Conducting the interview**

After the questions for the interview were drawn up the companies were contacted for appointments to conduct the interviews. The interviewer aimed to schedule the interviews at times that would be most convenient for the interviewees. After the appointments were made with the company, a short summary of the main topics of the interview as well as an estimate of how long the interviews will take were sent. In conducting the interviews the interviewer aimed to follow the guidelines as stated by Mathers, Fox, and Hunn (1998). These guidelines are discussed below.

When starting, the interviewer should take time to establish a rapport with the respondent. This can be done by explaining the purpose of the study and what will be done with the data gathered in the interview. Questions should be asked in such a way that it feels more like a conversation than an interview. Some other issues that were taken into account were (Mathers, Fox, & Hunn, 1998, p. 13):

- “Be familiar with aims and objectives of research
- Know your topic guide well; you may not get a chance to refer to it
- Record your interview if possible because you won’t be able to write it all down
- Reassure the respondent on the issue of confidentiality
- Be a good listener and don’t interrupt too much
- Try to start with factual background questions and move gently towards more specific personal questions
- Do not express your own personal opinions or appear biased - think in advance about your own prejudices, especially in the areas of sex, race, and age
- Use probes when answers need further clarification and respond to non-verbal cues
- Transcribe the tape as soon as possible after the interview. Never under-estimate the amount of time required for transcribing the tape and carrying out the analysis. A general rule of thumb is that for every hour of interview you have carried out, you will need to allow ten hours for the transcribing and analysis process”.

Because of the exploratory nature of the case study, it was decided to record the interviews on a laptop computer; this meant that the interviewer could focus entirely on the responses given by the interviewees and follow up interesting points with appropriate exploratory questions. Transcription of the interviews was done immediately after the interview was completed.

## **3.6 Data Coding**

The data gathered during the data collection method was then coded. Many computerized tools have been developed for this purpose. Two of the most well known tools are NUDIST and ATLAS.ti.

Even the use of such tools has advantages and disadvantages.

Glaser (1998) believes that the use of such tools:

- Introduces time consuming learning curves
- Inhibits development skills
- Hinders the researcher's creativity.

Others, like Fernández (2004) do not believe that using these types of tools has any negative effect on the coding process.

The author of this study made use of such a tool, in this case ATLAS.ti. It was the most helpful and easy to use. Fernández (2004) also makes a strong case for ATLAS.ti, who found the software intuitive, easy to use, efficient and full of useful functions.

## **3.7 Data Analysis**

The data gathered during the interviews must now be coded and analyzed.

According to Rubin and Rubin (1995), the purpose of the data analysis is to organize the interviews to present a narrative that explains what happened or provide a description of the norms and values that underlie natural behaviour.

Analyzing qualitative data is considered to be very difficult (Kohn, 1997). A big part of this difficulty can be attributed to the sheer amount of information that has to be dealt with. Therefore, techniques are required to organise and review these large amounts of information. Two methods that are used for data analysis viz. the constant comparison method and cross case analysis will be discussed.

### **3.7.1 Constant Comparison**

The constant comparison method can be used regardless of whether the data collected comes from one setting or from many settings.

The first step is to construct propositions from the data acquired in the first interview. Now, the next round of interviews or observations are done, and in so doing, the feasibility of the original

propositions are checked. If necessary, the propositions are updated. Because later interviews or observations might produce codes that were not obvious during the first round of coding, the researcher often needs to return to the previous cases to update the codes, and see if any new information becomes available. This is done until no more new codes are generated, or until theoretical saturation (Strauss & Corbin, 1990) has been reached.

The constant comparison method is mostly used in grounded theory and was thus not used in this study.

### **3.7.2 Cross Case Analysis**

If it is possible to divide the data into cases, then the researcher can use cross case analysis.

One of the main ideas in using cross case analysis is looking at data in many different ways (Eisenhardt, 1989). This helps to keep investigators from reaching premature conclusions (Soy, 1996). When using cross case analysis, the researcher divides the data of all the cases into type groups. These groups from the different cases are then compared to look for any similarities or differences. The findings are strengthened by any patterns that emerge from the data, while conflicts require the investigator to identify the cause or source of conflict. Analytic conclusions should be produced from the evidence that answer the original research questions (Soy, 1996). In the analysis of multiple cases, the researcher would usually start off by comparing only two cases, as if they were the only two available (Seaman, 1999). Comparing these will then lead to the formulation of propositions, with the data of the first two cases supporting the propositions. A third case is then analyzed to determine if it supports the propositions previously made. If this is indeed the case, this third case would then be added to the list of evidence, while a contradiction would cause the investigator to update the proposition accordingly.

### **3.7.3 Data analysis method used in this study**

The cross case analysis method was used as the data analysis method in this study. To assist in the task of analyzing the data, use was made of ATLAS.ti, a computer programme designed for coding and analyzing qualitative data. Other software programmes like NUDIST could also be used. The reason for using ATLAS.ti was mainly because of its availability. For a full comparison between NUDIST and ATLAS.ti see Barry (1998).

In the following paragraphs how the data coding process and cross case analysis were done will be discussed.



The data coding was done using ATLAS.ti. After all the interviews were completed the transcribed, interviews were stored inside a project created on ATLAS.ti. The project file is also called a Hermeneutic Unit (HU), and it serves as a container for the data, findings and ideas associated with a specific study. Once the transcription documents are placed inside the HU, they are known as Primary Documents (PD).

The primary documents were then read and codes were assigned to parts of the text that were of interest to the study. Any additional ideas concerning a specific piece of text or code were immediately stored as a memo on the programme, which could be linked to either a piece of text or a specific code. As the author worked through the PD's, more codes were added. This led to an iterative process, in which the author had to work through the PD's numerous times, each time updating the text with any new or updated codes.

After the coding of the texts was completed, the different parts of the documents that were assigned to a specific code could be compared. The comparison was simplified by the built in functionality of the ATLAS.ti programme which allows the codes to be grouped into families which can then be represented as a graphical network of all the associated codes, families and memos. As the codes were created and arranged into families, findings began to emerge, which were then further explored. The final step was to compare the coded data by means of the cross-case analysis method.

To perform the cross-case analysis the codes are presented as a matrix divided into the three cases as columns and the codes as rows. By means of this matrix view of the data, the cases can be easily compared to one another concerning a specific code. To deduct propositions from the data the method of comparing the first two cases was followed. Any differences or similarities were written up as propositions as if these two cases were the only ones in the study. After this the third case was then compared to the proposition created previously. Depending on the result of the comparison, the proposition was then modified. The cross-case matrix used in this analysis is given in appendix A.

The columns of the matrix are divided by the three distinct cases, being the different companies at which the interviews were done. Down the left column are the different subjects that emerged as dominant factors in the interviews and are used to compare between the different cases.

The cells of the matrix contain data gathered from the cases that apply to the specific subject.

### **3.8 Conclusion**

In this chapter different paradigms, approaches and methods available when conducting a research study were discussed. The specific methods that were used during this study were also highlighted and reasons given for their use. In this study, interpretive qualitative research was done utilizing the case-study method. Data was gathered by means of interviews which were then analyzed by using cross-case analysis.

In the next chapter a report of the case study will be presented. This report will display the findings of the data gathered during the interviews and analyzed by means of the cross-case analysis method.

## **CHAPTER 4**

### **4. RESULTS OF CASE STUDIES AND FINDINGS**

#### **4.1 Introduction**

To obtain a deeper understanding of the ways in which mobile telecommunications software systems are developed in practice, case studies were done at three different mobile telecommunication software development companies in South Africa. Data were gathered by means of interviews with project managers and developers. The questions asked in the interviews were determined beforehand and were designed to be open ended question on which the interviewees could easily elaborate. Because of the exploratory nature of the case study, it was descided to record the interviews on a laptop computer. This meant that the interviewer could focus entirely on the responses given by the interviewees and follow up interesting points with appropriate exploratory questions. The interviews were imediately transcribed after they were completed. After all the interviews were performed and transcribed, the data coding was done with the help of the software programme ATLAS.ti. The cross-case analysis method was then used to compare the data from the different cases with one another and derive valid propositions.

This chapter will be arranged as follows. First the research aim and objectives of the study will be reviewed after which a description of each of the cases as well as how and with whom the interviews were performed at each company will be given. This will be followed by individual discussion of the results obtained from each of the companies.

After the separate discussion of each company's results, the author will explain how the cross-case analysis was performed. The result of the cross-case analysis will then be given after which the list of development needs discussed at the end of Chapter 2 will be reviewed and updated.

#### **4.2 Research aims and objectives**

While conducting the case study the author kept in mind the original research questions of the study. These were to:

- Identify the systems development methodologies used (if any) by companies in South Africa in the development of telecommunication systems
- Highlight certain key aspects of mobile telecommunication software development

- Identify a list of necessities a systems development methodology should contain when used to develop mobile telecommunications software.

The author tried to determine, by means of open-ended questions, not only the steps followed by the companies during the development of mobile telecommunication systems, but also why these steps were followed. In doing so deeper insight into the possible processes, techniques and tools that might be required in an ideal mobile telecommunications systems development methodology was gained.

### **4.3 Results**

For this case study, in depth interviews were conducted at three different companies that were chosen as representative of the different mobile telecommunications systems available in South Africa. The issue of size difference led to one big company, one medium company and one small company being interviewed. All of the companies also developed distinctly different types of mobile telecommunication systems. Before the interviews were held, the author researched and went through any documents and news articles that could be found about the companies. This helped to give an increased insight into the culture and history of the companies at which the interviews were done.

In this section the findings from each case will be discussed separately from the other cases, focusing on two of the original research questions being: systems development methodology use and aspects of mobile telecommunication software development.

The discussion of each case will begin by giving a description of the company as well as how and with whom the interviews were conducted. The companies were assured that they would remain anonymous, and as such the companies' real names would not be used.

After the case description, the next part will be "systems development methodology used". Here the company's use of and thoughts about using systems development methodologies during the development process will be discussed. After this the systems development methodology of each company used during the development process will be presented by breaking the methodology down into the components discussed in the definition of a systems development methodology in Chapter 2. This seeks to answer the original research question of: "How are mobile telecommunication systems developed in practice?"

The second part of the discussion will be about the “Key aspects of mobile telecommunications software development”. Under this heading several key factors identified during the interviews and subsequent analysis will be discussed. These will be ordered as follows: Firstly information will be given concerning the “Flexibility and Adaptability” of the developed systems, the systems development methodology followed as well as the company itself. Secondly the issue of development time, which played an important part in telecommunications system development, will be discussed. This will be followed by, how the company and systems developed have to communicate with other systems. Finally any issues and ideas inherent in the telecommunications industry will be identified. During the discussion the notation “(code)” will be used to reference a specific quote or idea back to the original code in the transcription.

#### **4.3.1 Case 1: Company 1**

##### **4.3.1.1 Company description**

Company 1 is a big mobile telecommunications service provider and software development centre. Their system development team, compared to the size of the company, is relatively small consisting of 8 members.

They develop a wide range of different types of telecommunications systems, from customer management and billing systems to customer value added services with GPRS and more recently 3G. Many of the systems they develop are sourced internally, though they also build systems that are requested by the major mobile networks in South Africa (being Vodacom, MTN and CellC).

The history of Company 1 goes as far back as the late 1980’s when they entered the paging market. They grew quickly and within a few years enter the GSM digital cellular market and started a South African software development centre. After this they soon rose to be South Africa’s largest independent dual service provider.

The interview at Company 1 was done at their headquarters in Midrand where their IT department is situated. The company allowed the researcher to conduct only one interview, stating that their department was extremely busy as that time. At Company 1 it was decided to conduct the in depth interview with a project manager, who also had experience as a mobile telecommunications system developer. This increased the depth of the interview by drawing on the interviewee’s experience in both fields. The interview was held in the interviewee’s office and took about two hours. The interviewee was first given a brief description about the study as well as what the general aim of the

interview would be. Following this the interviewee immediately explained the process the company followed when developing a new system. This was followed up by questions to answer any issues the interviewer felt needed to be explained in more detail, as well as the interview questions that were not answered during the discussion. During the analysis of the study, the interviewee was contacted telephonically to follow up on any information that was felt to be unclear or insufficient. After the analysis of the case studies, the propositions and findings were sent to the interviewer via email to determine whether the interviewee felt that the results were correct.

The interviewee had six years experience in the mobile telecommunications industry, three and a half of which had been as a project manager at Company 1. Before that he was a developer at another telecommunications company.

#### **4.3.1.2 Systems development methodology**

##### ***4.3.1.2.1 Systems development methodology used***

When developing new systems the company did not use a formally developed systems development methodology, as those discussed in the first part of Chapter 2. One of the main reasons for this was because the interviewee believed that using a formal approach increases the time taken to develop a system (87). They followed an in-house developed systems development methodology. This process ensures that everything is taken into account, the system functions correctly and the development process is better controlled (32). Another aim of the in-house systems development methodology is to keep the development process as simple as possible. This simplicity has the added advantage of decreasing the development time (108). Although a formal systems development methodology was not used during the actual development of the project, the PMBOK methodology was used to facilitate the management of the project (97).

#### **Approach**

The company followed a mixed approach when defining the systems development methodology used. Speed of development was given as a key goal, while modular development, component reuse and architecture centric design is based on object-oriented principals.

#### **Process Model**

The development follows a linear style, with some modular development that could be done in iterations.

## Method

### 1. Feasibility Study

If it is a big project a feasibility study is done to determine the ROI. This study takes into account, development time and also hardware purchases (61).

### 2. User Requirements

Gathering the user requirements is a very important part of the development process. Not only are the requirements necessary to develop the system, but the user requirements are also used at the end of the development process to determine whether the project is successful. “Because at the end you could say, here are the user requirements. And you measure the end product against it” (56).

A project starts off with a user sending a requirements specification to the project manager. This requirements document is often incomplete and should first be extended before the system can be built. “It does not help that they give me one sentence, and with that one sentence I have to develop a project” (10). Extending the user requirements establishes exactly what the user wants the system to do; it also aims to capture the user’s thoughts and thought patterns on paper. Often the users have a current manual or outdated system in place, which should also be documented. All this information is then put into a user requirements specification document which is reviewed by the project manager and developers before development continues (69). If this document is thought to be insufficient, the user requirement specification is converted to a “Functional Specification” (70).

It does happen that sometimes development continues before the user requirements gathering and analysis phase has been completed (“from that one sentence one has to make a project,” 29) and although developing a system without complete user requirements is possible and some time is saved by not having to wait for user feedback, developing a system with incomplete requirements is very challenging (29).

In the interview Company 1 stated that gathering the user requirements was sometimes difficult and time consuming, but that it is worthwhile to spend some extra time on this task. “It sometimes takes a while, but then I know for a fact that the end product we deliver is what everybody agreed to” (87).

### 3. Design

After the user requirements have been documented in a user requirements specification document or functional specification document, the specification is then circulated to all the departments that will be involved in development (66). They look at how the new system will influence the current systems within their department (86) and the business rules in their department that the new system must adhere to (67). After all this information has been gathered they have a complete specification (68).

The system is then modeled using a **flow diagram** which shows the key parts of the system. It also shows how the system will interact with other systems, especially with the main cellular networks (being Vodacom, MTN and CellC) (12).

Depending on the size of the project the system is then divided into modular pieces (91). This modular development makes the management of the development process much easier (73, 91). During this phase it is also determined whether there are any pre developed applications which can be reused (30). An accurate estimate is then calculated for how long the development will take (90) which is important from a project management viewpoint.

Buying and the setup of hardware for the system to run on should also be addressed during the design phase (61).

### 4. Development

After the system has been designed and broken down into modular parts the development of the system can begin. The interviewee stated that control and management of the development process was important (73) and that he tried to have short progress meetings at least every two days (74). These progress meetings show whether the development is still on the right track as well as in the given time frame (74). A few of the key elements during the development process will also be discussed, these being the reuse of existing code and the development languages used.

#### **Reuse**

The reuse of ready developed code was an important factor during the development of any system. The developers would always first check whether an application had been developed previously with the required functionality. A small change could then be made to this



application or code part to give the required result. “Isn't there already something in place that we can just change, that still gives the same result?” (30).

### **Development Languages**

Company 1 mostly works with Microsoft products, doing most of their programming in Microsoft dot Net (51). Some of the other languages like Delphi and HTML are also used, though not as much (51).

#### **5. Testing**

After the system has been developed and the different modular parts have been combined, the system can now be tested. Some elements of testing are to ensure that the system “can communicate with the networks” (75) and also to assure the quality of the system (57). If the system will influence many people stress testing is done to ensure that the system can handle big volumes (76). After the system’s communication abilities, quality and load handling capabilities have been tested it can then be released to the end user for user acceptance testing (UAT) “You release to the users for user acceptance testing” (75). Depending on the outcome of the UAT, the product can then be released.

#### **6. Backup and deploy**

Before launching the product, a secure backup system is created, together with a backup strategy in case the product fails (42, 46). Once all this is in place, the system can be released.

#### **7. Support and Maintenance**

After the release of the system, maintenance is done to keep the system up and running and thus keep the client happy “Because if something goes wrong, the client becomes unhappy” (93). Any errors that are discovered by the users or maintenance personnel are logged immediately. They are then analyzed to see whether they are critical errors. Critical errors are addressed immediately “If it is critical the programmer should look at the problem immediately” (79), while less important errors are addressed during maintenance days, which is every Friday.

One of the problems the company faces during support and maintenance is that maintenance takes some of the resources away from development, in that the developer that worked on

the original project has to drop every thing to do maintenance on the system (see Challenges) (82). The interviewee suggested that the department should be restructured to contain personnel who focus specifically on maintenance, thus freeing up developers to focus only on new developments (82).

### **Techniques**

If the size of the system is very large or the system is particularly complex, the system will be divided into modular pieces and developed in a modular fashion (73, 91). The identification and reuse of pre-developed components was also an element of every system that was developed (30).

- **Documents Used**

- **Specification Document**

One document that was used during the development process at Company 1 was a general specification document. This document is created following user requirements analysis and then sent back to the user for approval (10). The specification document forms the basis from which the system is developed. It contains business rules, data flows and user requirements, everything the user needs (68). This document also serves as a means of communicating the project information throughout the business. “When we get the specification document, we distribute it to all the departments” (69). If the specification document is thought to be incomplete (or unclear) it is extended to a functional specification. This functional specification goes into more detail about what the system should do and its relationship to other systems. “Then you go into more detail” (71) “Although this product has to do the following, what relationship is it going to have with other systems?” (71)

After the specification or functional specification document is done, it is used to create a flow diagram (12).

- **Flow Diagram**

The user specification can be used to create a flow diagram. “What I do with the specification is I draw a flow diagram” (12). This document shows how the system interacts with other systems in a much more logical and readable way.

- **Tools Used**

Whether company 1 used tools depended on the project “It depends on the project” (28). The tools that they used were mostly project management tools, and these tools were only used for big projects. “If the project is very big we’ll use Project Manager” (28).

#### **4.3.1.3 Key aspects of mobile telecommunication software development**

##### **4.3.1.3.1 Flexibility and Adaptability**

Because the company’s systems are in constant communication with the main networks (Vodacom, MTN and CellC), any changes introduced by these networks have to be incorporated in the company’s current systems (23). The company systems are also governed by national telecommunication standards. Thus a change in standards, or a required compliance introduced by the networks (Vodacom, MTN and CellC) or ICASA (Independent Communications Authority of South Africa) has the ability to influence all the systems and business rules in a company (23). For this reason the company and systems need to be flexible enough so that they can be updated quickly when the need arises (23). Another area where flexibility is needed during the development process is to handle any unforeseen events. “Today everything works fine, tomorrow nothing works anymore” (83). These unforeseen events were mostly caused by issues with previously built systems, which then first had to be resolved before normal development could continue.

##### **4.3.1.3.2 Development Time**

One strong aspect of telecommunications system development in Company 1 is that the time available for development is very short, “It has to happen very fast” (14). Many times there are only a few weeks available in which to develop an entire system, “Two to three weeks” (7). This is one of the reasons why no formal systems development methodology was used during development, because according to Company 1, using formal systems development methodologies increased the time it takes to develop a system, “Sometimes it takes a long time” (87).

##### **Reason:**

One of the reasons why the time available for telecommunication development is so short is because of pressure from the main networks to deliver cutting edge products. “The networks want that cutting-edge on the market, so they remain silent with a project right up to the last moment” (7).

Another reason is the rapid growth in technology (96), as well as the expansion of the telecommunications industry in South Africa (14, 18, 25).

#### **Obstacles:**

The time available for development can be decreased by many unforeseen circumstances. “Something goes wrong with a previous project” (83). What often happens then is that developers have to stop what they are busy with and do maintenance on another project. “Then that guy must do maintenance” (83).

#### **4.3.1.3.3 *Communication with other systems***

All the systems at Company 1 were in constant communication with other systems and networks. This means that each system has to be synchronized with 2 or more systems all the time. “We are always linked to Vodacom, MTN and CellC” (13). During the development process this also has to be taken into account. While designing a system, it should be designed in such a way as to ensure that the system can correctly connect to the required systems (12). System integration also plays a role after development. When testing the newly developed system it has to be tested whether the system correctly interfaces with the other networks before launch. “If I am pleased with the testing, that it can communicate with the other networks...” (75).

This constant interaction increases the complexity (71) when developing a new system and brings with it numerous challenges. Company 1 has many systems that are currently in production, most of which are influenced in some way by one or more other systems. This means that when a new system is introduced it should be established how the existing systems will be affected, and these systems would then have to be updated accordingly (2, 23). It also means that an industry change, like number portability has an effect on every single system running within the company. “...one thing, like number portability influences all our systems.” (23).

#### **4.3.1.3.4 *Environment***

Company 1 described the telecommunications industry as a very challenging industry to work in. “It is a very challenging environment” (65). Some characteristics of the telecommunications industry, which were identified during the interviews, were that the industry is controlled by many governing bodies, like ICASA (22), and that in the telecoms industry, security is very important (53).

#### **4.3.1.3.5      *Security***

Security is a very important issue in all of IT (41) and especially in the telecommunications industry where access control is very important (53). Security is even more important when dealing with big clients that utilize big systems (41). If for some reason security has not been properly applied, it could cost the company much money (54).

#### **4.3.1.3.6      *Backup Strategy***

Before launching the product, a secure backup system is created, together with a backup strategy in case the product fails (42, 46). Setting up this backup strategy and making sure the backup is complete and virus free is challenging.

### **4.3.2 Case 2: Company 2**

#### **4.3.2.1 Company Description**

Company 2 is a medium sized company with about 30 employees in each of its three departments. There are four members in the development team. The systems developed by Company 2 are not limited to mobile telecommunication systems. The telecommunications systems they develop though focuses more on mobile entertainment services (like ringtones, logos, picture messages, etc.), incorporating SMS, MMS and web based “WAP” services (including m-commerce systems). The company has been developing mobile telecommunication systems for seven years and can be credited with launching South Africa’s first WAP portal and could be credited with being one of the largest mobile entertainment service providers in South Africa.

The interview at Company 2 was conducted at their headquarters in Fourways. Contact was made with the IT department’s lead developer and project manager who also volunteered to be the interviewee. As was the case with the first company, the company was extremely busy and could allow only one interview and asked that it be no longer than one hour. An in depth interview was held in a secluded conference room with one person who fulfilled two roles, being lead developer and project manager. The interviewee was first given a brief description of what the study was about as well as what the general aim of the interview was. The researcher then continued with the rest of the interview questions and tried to get the interviewee to elaborate on certain issues. During analysis, as with Company 1, the interviewee was contacted telephonically and via email, and asked to elaborate on any issues that the researcher found to be unclear. The results and propositions were emailed to the interviewee for confirmation and feedback purposes.

The interviewee had, until the time of the interview, five years experience in the mobile telecommunications domain and has worked at Company 2 as a software developer since graduating, and later as project manager.

#### **4.3.2.2 Systems development methodology**

##### **4.3.2.2.1 Systems development methodology used**

Company 2 used an in-house developed systems development methodology (“Its something we developed ourselves” (14)), which they tried to follow as much as possible (16). This systems development methodology was designed to give structure to the development process while cultivating a relaxed atmosphere, which leads to better productivity. “I feel people are more productive if the environment is more relaxed and informal” (20). To what degree the company followed this systems development methodology depended on the size and type of the project. At Company 2, the systems development methodology was especially useful when developing bigger projects that work with sensitive data, because it makes sure that everything is taken into account and the system functions correctly. “You try to apply it as thoroughly as possible so that it can be done properly” (20). The interviewees’ view about formal systems development methodologies was that they worked in theory, but not in practice (“What it says in theory and how it works in practice is not always the same” (16)), and that only big companies really used formal systems development methodologies (19). They also recalled that in their experience, bigger companies that have many strict procedures to follow, “takes forever” (18) to get things done.

#### **Approach**

The main approach for this systems development methodology is speed, while some object-oriented techniques were used in order to increase the speed of development.

#### **Process Model**

A linear process model was followed with some possible iterative coding during modular development.

## **Method**

### **1. User Requirements**

Gathering the user requirements is a simple process and no strict procedures are followed during this phase. “It is very informal” (14).

What generally happens is that the company gets the user requirements document, then using their knowledge of the domain sets up a technical document or specification which specifically applies to their network infrastructure (14). This document is then sent back to the user for approval. “Get it approved by everybody” (12). Once this specification is returned the document is then used as a base for development. Severe time constraints often cause this step to be neglected somewhat. “The whole procedure of getting the requirements and setting up the spec and that sort of thing... There is not always time” (18).

### **2. Design**

The interviewee elaborated very little on the subject of planning and design. In fact it seems that design phase is often disregarded because of the time constraints placed on development. “There is not always time to go through the whole thing of planning or product testing” (40). The design phase might indeed be totally unnecessary because Company 2 has a development platform setup to develop systems faster and more efficiently (See Development).

### **3. Development**

During development, speed is an essential element. Because of this Company 2 tries to stay away from strict development procedures because they tend to take up much time (18). In the interview it was specifically stated that most big companies they work with used strict procedures during systems development. “So they have a much more strict procedure for how everything works, and it takes forever” (18).

## **Development platform and Reuse**

One of the elements that played a major part during the development process was the fact that they reused many pre-developed code and system parts (26, 27). “We try to reuse as far as we go” (26).

Company 2 has a set of ready-made development components and pre-developed code that they reuse. “A whole set of tools that we developed ourselves” (27). These are standard components that are required in every system that the company develops. Some examples of

these are database interfaces or SMS interfaces (27). This means that if the system they have to develop needs to send an SMS all the functionality they need is already available in a pre developed interface. Maybe only one line of code needs to be changed. “Rather than rewriting the functionality you need you just need to change one line of code” (27).

### **Development Languages**

Mostly the systems developed by Company 2 are coded using the Microsoft dot Net framework with C# being the main development language (8). The databases used were also Microsoft products (8). Visual Basic was also sometimes used (7).

#### **4. Testing**

Although testing is an essential part of developing a new system (28), the time constraint placed on development does not always allow the product to be extensively tested. “Because of the speed of development you need to build the system and fix any problems afterwards” (40). Testing is only done right before the product is released (12, 18, 30) to ensure the quality of the product (28). According to Company 2, they “don’t test from a development aspect, but from an operational aspect” (29), so after the system is released, it is monitored and tested constantly by maintenance personnel to ensure that the system functions as it should. “We test for uptime, this is our main reason for testing” (29).

#### **5. Support and Maintenance**

Company 2 also has people who monitor the system 24 hours a day after the system has been rolled out and in addition to errors logged by users, also do their own tests on the system (28). This means that if a problem is found, they can respond to the error within minutes (28). Updating the system is done quickly, without going through any strict procedures (30).

### **Techniques**

- **Documents Used**

- **Specification Document**

This document is created following user requirements analysis (“Typically we’ll get the requirements... We will then setup the specification” (14) and then send back to the user for approval. “Get it approved by everybody” (14). The specification document takes into account how the internal network of the business works and how the system will fit



into that network. “We will then setup the specification of what is necessary, and more specifically, what is relevant to our systems” (14).

- **Tools Used**

Company 2 mostly made use of Microsoft products during development. “We use 99% Microsoft product” (8).

#### **4.3.2.3 Key aspects of mobile telecommunication software development**

##### ***4.3.2.3.1 Flexibility and Adaptability***

According to Company 2, the development process must be kept flexible and adaptable. “You try to keep the systems on your side as open as possible” (24). This is because you are always working with other development teams, and the systems being developed are constantly communicating with other systems (24, 26). Part of Company 2's strategy for keeping a system flexible, is to reuse ready developed code. “We try to reuse as far as we go. This fits in with keeping the system as flexible as possible” (26).

##### ***4.3.2.3.2 Development Time***

Because of the severe competition in the telecommunications industry, “cut throat” (10) systems need to be developed in very little time (10). Because of this, there is not always time to draw up accurate specifications, or to do proper testing (10). Company 2 believes that only big companies need to use strict procedures. “Big companies have lots of strict procedures” (18). Using formal systems development methodologies increases the time it takes to develop a system. “...takes a month. What could have taken a few days” (19).

##### **Decreasing development time:**

Being flexible and not having strict procedures could help bring down the time it takes to develop a system. Company 2 shared this view by saying that “big companies have lots of strict procedures, and the thing is... it takes forever” (18). In order to decrease the time it takes to develop a new system, Company 2 tried not to use strict procedures, thus creating a more informal, productive development environment (20). Another way for the company to reduce development times was to reuse existing code and system interfaces (26, 27).

#### **4.3.2.3.3      *Communication with other systems***

Every system the company builds has to be in communication with at least one of the big cellular communications networks (Vodacom, MTN or CellC). “We are dependent on these guys” (24). The problem with being dependent on other networks is that sometimes these companies are much slower and less adaptable. “Because they are big companies their service is not always 100%. You wait a long time for them to get things done on their side” (22). To minimize the negative impact communicating with other systems would have on development, Company 2 kept their systems open and adaptable to be able to fit in with other networks. “You try to keep the system on your side as open as possible” (24).

#### **4.3.2.3.4      *Environment***

Even though compared to other countries, telecommunications in South Africa is very immature (3), it is a fact that telecommunications, as a technology, has become an absolute necessity in South Africa and the world. “The world cannot function without telecommunications” (31). Company 2 stated that the telecommunications industry is “one of the most difficult industries to work in” (31). Telecommunications is a “cut throat” market, where everything has to happen very quickly (10). Also, companies have to conform to the rules and regulations, as determined by the major networks (Vodacom, MTN and CellC) and governing bodies (eg. ICASA) (24), and they can sometimes be difficult to work with (22). These rules and regulations determine how telecommunication systems should function as well as any services these systems should take into account, they do not determine how these systems are developed. This is a problem, because according to the interviewee there are very few, if any, strict international standards concerning the protocols which should be used in systems “There are no strict standards in use regarding protocols” (23). Most protocols are just built in-house (23). One good thing about the telecommunications industry in South Africa is that it is a good industry to be in because it is constantly growing (35).

### **4.3.3      *Case 3: Company 3***

#### **4.3.3.1      *Company Description***

Company 3 is a very small company with only 10 employees. The company was established in 2001 and focused on creating GSM-based products like messaging, USSD applications and VAS systems. Not only do they develop mobile telecommunication software, but they are also involved in designing and building hardware telemetry systems. The types of mobile telecommunications

systems they develop are at a much lower level than the other two companies, extending from data terminals down to routing and billing systems in the GSM network itself. Billions of packets of data are routed at a very high speed from the customer to one or more content providers and back, during which time the data is transformed, filtered and aggregated. The company has also won numerous awards at the Technology Top 100 competition including; most promising startup company; Technology 100 Company twice; the award for Information, Communication and Technology: Hardware; and most recently they were a Technology Top 100 Innovation Finalist.

The interview at Company 3 was conducted at their headquarters in Centurion. Contact was made with the company's lead developer and the company was immediately very eager to help with the study. In depth interviews were conducted with two employees at Company 3; one was the lead systems engineer and developer, while the other was a project manager as well as the managing director of the company. It was decided that both interviewees should be interviewed at the same time. Having two interviewees in one interview had the advantage that one interviewee could easily pick up on something that the other might have missed. One could also elaborate on a point that the other had made. The interview was held in the company's conference room and took about two hours. The interviewees were first given a brief description of the study was about as well as what the general aim of the interview would be. The researcher then continued with the rest of the interview questions and tried to get the interviewees to elaborate on certain issues.

Contact was maintained with the interviewees during the analysis of the interview, and they were asked to clarify and elaborate on any issues the researcher found vague or incomplete. The two interviewees were also emailed the final results and propositions so that they could comment on these and see whether they agreed.

#### **4.3.3.2 Systems development methodology used**

##### ***4.3.3.2.1 Systems development methodology***

Company 3 developed their own systems development methodology (26) which they tried to use whenever the available development time allowed it (14). This systems development methodology was specifically developed to be compatible with their business culture and character. "You must develop your own methodology that fits in with the character of your business" (27). Because of the young, dynamic and informal culture of the company, the in-house systems development methodology they developed was not static, but was able to be changed and adapted for every client

they worked with. “It changes to adapt to each client. Each client has a different way of working” (33). The company did not always follow this systems development methodology when developing new systems, but the interviewees conceded that when it was used it had a positive influence on project development. “The times that we do it like that, it is always successful” (14). More formal systems development methodologies, like those discussed at the beginning of Chapter 1, did not fit in with the informal nature of the company and were thought to be better suited for big companies (27).

### **Approach**

The company aimed to keep the development process as flexible and adaptable as possible. The systems development methodology used was not static, but could be changed and adapted for every client they worked with. Certain techniques were used that were common in object-oriented development, these being the identification of and reuse of components, the architecture centric design of the system, as well as the modular development style.

### **Process Model**

A linear process model was followed with iterative coding and testing during modular development.

### **Method**

#### **1. User Requirements**

Before any system can be built, one must first know what the customer wants. Getting proper user requirements can be, as Company 3 put it, “Painful” (12). The way in which Company 3 usually gets the user requirements is that they have the user write down what he wants. That is then translated into a technical document which is applied to their development platform (17) and sent back to the user for approval (13). In an ideal world one would wait until the document is returned before development starts (18), but this takes too long (“2 to 3 weeks” (18)). What really happens is that by the time the user accepts and sends back the specification, the system is basically finished (18) and any adjustment to the system is made afterwards. “Once the signed off document is returned... We will already be done with development” (18). The developers at Company 3 are aware that there is “always a trade off” (28) between the time saved during the development of a system and the time that will be spent debugging the system afterwards. A challenge that comes in during the user requirements gathering phase, and remains an issue throughout the project, is managing

customer expectations (41) and ensuring that the idea the customer has in his head, is what he is actually going to get.

## 2. Design

Using the specification document, as well as the knowledge of the company's development platform, the system is divided into modular pieces (9). Some applications could consist of thousands of small applications all communicating with each other. "It's a thousand small programmes that sit there and communicate with each other" (11). Furthermore, these applications can be distributed across several physical machines. "They don't have to be running on the same machine" (11). All of these issues have to be taken into account when designing the system.

## 3. Development

The actual coding of the system is not the most difficult part of the project (40). This is because while most other parts of developing a system might change from project to project, the generation of code always stays the same (34). The system is developed in parts, each modular part is developed and tested, and after the development all the parts are then combined to form the entire system (31).

### **Development platform and Reuse**

In the interview with Company 3 it emerged that they make use of a development platform when developing their systems. This platform consists of a library of reusable applications that they call "The Engine" (9, 11, 17, 25). When the client wants a new system developed, the ready made applications are chosen from the library, modified for the client's specific needs and used to build the new system. "You have a basic platform you use, then you make small changes on it" (8). The platform has to be very stable (8). This means the applications in this library should have already been properly tested and made certain that they work correctly. Not only does reusing these applications reduce the development time, but it also simplifies the development process because it promotes modular development. "The way the engine is built is modular; it is divided into functional groups" (9). "Typically you just go and change some of the interfaces, and a bit of the functionality, but you never rewrite all of it" (9).

### **Development Languages**

The main languages used when developing a system at Company 3 are Java, C, XML and XXL (48). For developing “non core” (50) applications that were not crucial to the functionality of the system, languages like Delphi or Object Pascal might be used; this includes applications like user interfaces and some graphical tools. “We use Delphi, with object Pascal for client side applications” (C3).

#### **4. Testing**

Each modular part of the system is first developed and tested separately. After the development all the parts are then combined and tested as a complete system (31). To test the system it is loaded into a testing environment. This environment is safe and controllable. Here the system is tested and fixed before it is released. “Nothing goes to the live environment before it has been signed off on the testing environment” (25). Simulator tools are especially useful for testing the system. Company 3 would write a simulator that functions exactly the same as the network or system the new product would have to communicate with (32). This means that instead of having to wait for feedback from the other system to say whether the product works correctly, the product can be tested on the simulator. “We test the system completely, and ATP’s are signed off” (25). After these tests are successful, the system can be released into the live environment, where it once again goes through a bout of user testing (25).

#### **5. Support and Maintenance**

According to Company 3, the maintenance of telecommunication systems is critical. “It is mission critical. Each one of us in the company has been on twenty four hour standby for the last ten years” (5). Every project that is developed is sold with a support and maintenance contract. “In this industry you never sell just software. You sell software and then you also have support and maintenance and a service level agreement” (66). Company 3 has a central point where all the systems currently in production are constantly monitored (66). Site management and version controlling tools are used during development, and this makes maintaining the system much easier. “Version control. The Site management stuff... This makes support easier” (57). Other tools, like MRT cheap, are also used during maintenance to monitor system statistics and events (52).

## Techniques

- Documents Used

- Specification Document

The developers at Company 3 attributed their extreme flexibility partly to the fact that they used very little documentation (30). The company did however make use of two documents, the general specification document and a protocol specification document. The general specification document is created following user requirements analysis and then sent back to the user for approval. “Send it back to him that he signs off on it” (13). It takes into account how the internal network of the business works and how the system will fit into that network. “We translate the user requirements into a technical document and conform it to our engine and to our platform” (17). This document is also used as a basis for setting up the user acceptance testing, which is done after the project is finished (20).

The other specification document which is used by Company 3 is the protocol specification, which describes the way the system should interact with other systems. “That document shows what the protocol specification looks like. For the interface to whatever” (39).

- Protocol specification document

The protocols used by the system to communicate with the client systems are documented (36).

- Tools Used

Company 3 used tools during the development of a new system. Most of the tools used were developed in-house, while the few “outside” tools that were used were mostly Open Source. “Most of the tools we use are open source, or developed in-house” (52). These tools that were purchased outside the company were only used in “non-core” areas of development that are not necessary to the functionality of the system being built. Some of the tools Company 3 uses are simulator tools that are used for testing purposes. “Simulators, as a tool, work very well” (45), while tools are also used for management purposes, like code management and version controlling (57). Statistic tools and logging tools are also often used. “We use RRD tool en MRT cheap for our statistics” (52). Once the workings of the application go deeper down into the network, one moves out of the MS tool base completely. “When you walk out

of the door, you are in the Microsoft Windows world, when you walk in the doors, you are no longer in the Microsoft Windows world” (6).

#### **4.3.3.3 Key aspects of mobile telecommunication software development**

##### **4.3.3.3.1 Flexibility and Adaptability**

Company 3 declared that they were very flexible in their approach, and that this approach is very advantageous. “The advantage we have is flexibility, extreme, extreme flexibility” (30). The cause of this extreme flexibility is the fact that they are a small company with only ten employees. This, combined with the fact that they use little documentation allows any changes to be addressed very quickly. “Things can just happen” (30). Company 3 also focused on reuse, in this case having a whole library of pre-developed code that they could select to use in the system and modify where needed.

According to the interviewees, flexibility is a very important factor in mobile telecommunications system development. Systems development needs to be very flexible because companies are always working with other development teams, and the systems being developed are constantly communicating with other systems (30, 42).

##### **4.3.3.3.2 Development Time**

The time available to develop a telecommunications system is generally very short. “You’ve got five weeks, then they want to be up and running” (16). Sometimes the development time was no more than a few weeks (15, 16). Because of these short development times there is not always time to draw up accurate specifications, or to do proper testing. “In practice the documentation process is going to take, if you're lucky, two to three weeks” (18).

- **Reason**

One of the reasons why the time available for telecommunication development is so short is because of pressure from the main networks to deliver cutting edge products. “They search and search, and suddenly they go. Ok, we want your product and we want to go live at the end of next month.” (15).

- **Decreasing development time:**

Being flexible and not having strict procedures could help bring down the time it takes to develop a system. “...extreme flexibility... we can, very quickly, make it work exactly how



they want it” (30). The developers were aware that developing faster does come at a cost. This is because there is always a trade off between the speed at which you develop the system and the time it is going to take to debug the system. “Always a tradeoff.” (28).

- **Obstacles**

The time spent on communicating with customers and other project teams caused much time to be wasted. This contributes to increased development time. “The more you talk to the people the happier they are. But this takes more time” (46).

#### ***4.3.3.3 Communication with other systems***

Because the systems developed by Company 3 had to fit in with other networks, the systems had to be kept open and flexible. “Flexibility, extreme, extreme flexibility” (30). This connection with other networks also makes it necessary to keep track of the protocols used (“Those protocol specs must be kept up to date” (39)) and to make sure to test whether the system correctly interfaces with the other networks before launch.

The way in which the systems are developed must also be flexible and adaptable, because sometimes the system has to be developed in conjunction with development teams from other companies. The problem with this is that sometimes these companies are much slower and less adaptable than them. “We have to integrate with components developed by companies that are not as flexible as us” (42). This is a difficult problem to overcome. “There is really nothing you can do” (44).

#### ***4.3.3.4 Environment***

- **The Telecommunications Industry**

The telecommunications industry is very big and consists of many different facets, voice, data, customer call centers, SMS, MMS, USSD to name but a few. “It is a very big, multi faceted environment” (68). Telecoms have become such a big part of our daily lives that every system influences thousands, if not millions of people (5). And most of these systems are mission critical and have to be maintained constantly (5). The industry poses a particular challenge to new development companies, because of its high barrier of entry (69).

- **Challenges**

Company 3 mentioned the difficulties they experienced when communicating with outside networks (see Communications with other Systems). They said that a possible solution to this problem was to apply commercial reasons for them to speed up communication. “So they know they are not getting paid. Or if the project takes too long, they have to pay a penalty” (47). Another challenge that was mentioned by Company 3 was that of managing customer expectations (41). To ensure that the system the customer is going to get, is that which he is expecting. Two more challenges were gathering user requirements, which was said to be “Painful” (12) and the short time that is available for system development (15, 16).

#### **4.3.3.3.5 Security**

The types of software they develop do not have any real security issues (58).

However, the servers the applications run on should be secure (61).

## **4.4 Cross case analysis**

For this study the researcher chose to analyse the data gathered through interviews by using the cross-case analysis method. To start with, the data coding was done using ATLAS.ti. The transcribed interview documents were placed inside a project created on ATLAS.ti. The project file is also called a Hermeneutic Unit (HU), and it serves as a container for the data, findings and ideas associated with a specific study. Once the transcription documents are placed inside the HU, they are known as Primary Documents (PD).

The primary documents were then read and codes were assigned to parts of the text that the researcher deemed to be of interest to the study. Any additional ideas concerning a specific piece of text or code were immediately stored as a memo on the programme, which could be linked to either a piece of text or a specific code. As the researcher worked through the PD’s more codes were added. This led to an iterative process, in which the researcher had to work through the PD’s numerous times, each time updating the text with any new or updated codes.

After the coding of the texts was completed the different parts of the documents that were assigned to a specific code could be compared. The comparison was simplified by the built in functionality of the ATLAS.ti programme which allows the codes to be grouped into families which can then be

represented as a graphical network of all the associated codes, families and memos. As the codes were created and arranged into families, findings began to emerge, which were then further explored. The final step was to compare the coded data by means of the cross-case analysis method.

To perform the cross-case analysis the codes are presented as a matrix. The columns of the matrix are divided by the three distinct cases, being the different companies at which the interviews were conducted. Down the left column are the different subjects that emerged as dominant factors in the interviews and used to compare between the different cases. The cells of the matrix contain data gathered from the cases that apply to the specific subject.

By means of this matrix view of the data, the cases can be easily compared to one another concerning a specific code. To deduct propositions from the data the researcher followed the method of comparing the first two cases. Any differences or similarities were written up as propositions as if these two cases were the only ones in the study. After this the third case was then compared to the proposition created previously. Depending on the result of the comparison the proposition was then either modified or extended. The cross-case matrix used in this analysis is given in appendix A.

## **4.5 Cross Case Analysis Findings**

In this section the findings of the cross-case analysis will be reported. This description is ordered the same as the findings of the individual companies. First is the discussion about the systems development methodology used, after which a discussion on the elements present within the systems development methodologies follows. Secondly the discussion about the key aspects of mobile telecommunications software development is given. Lastly, the list of software development needs and elements of the telecommunication environment that were discussed in the last part of Chapter 2 is reviewed and updated.

### **4.5.1 Systems development methodology used**

#### **4.5.1.1 Systems development methodology use**

The companies interviewed used their own, in-house developed systems development methodologies.

There was a general feeling amongst all the interviewees that using a formal systems development methodology has a positive influence on project development. Although there is still much debate about this issue in academic circles, many believe it to be true. Avison and Fitzgerald (2002, p31)

state that “The use of a methodology improves the practice of information system development”. Some of the positive points made were that formal systems development methodologies ensure that the system takes everything into account and functions correctly. It also gives control to the development process. However it can detract from the relaxed development atmosphere, which is conducive to productivity. Not only this, but as stated above, it is a strong belief that using a formal systems development methodology slows down the development process.

A big factor determining the use of a systems development methodology is the size and type of the project. Bigger projects that work with sensitive data or money require a formal systems development methodology to be used to make certain everything is taken into account and the system functions correctly. This increases the time taken to develop the system. The systems development methodology used must be compatible with the business. The systems development methodology must be adaptable for every client. A systems development methodology that is adapted to fit different situations is also called a “contingency methodology” (Avison & Fitzgerald, 2002, p546).

Some criticism regarding the use of formal systems development methodologies during development was that systems development methodologies only work in theory, not in practice. Some also believe that only big companies are able to benefit from formal systems development methodologies.

### **Proposition 1**

The feeling from all three the companies was that using a proper systems development methodology was beneficial to the development process, because of the added control and decreased development time it provides. In-house developed systems development methodologies that could adapt to the project needs were mainly used, while more formal rigid systems development methodologies were thought to be unsuitable and too theoretical.

#### **4.5.1.2 Process Model**

All three the companies utilized some techniques which are often used in object-oriented development and these were the identification of and reuse of components, the architecture centric design of the system as well as the modular development style. The main approaches at two of the companies were on development speed, while the other company focused more on a flexible and adaptable approach.

#### **4.5.1.3 Approach**

All of the companies followed a linear style of development with some possible iteration during the code generation phase

#### **4.5.1.4 Method**

Here some of the phases that were followed by all the companies will be discussed.

- **User Requirements**

Before any system can be built one must first know what the customer wants. Getting proper user requirements can be, as Company 3 stated, “Painful.” One might need to get more out of the customer because some times all the user gives is a document with one sentence, stating what he wants the product to do. A common solution to this is to get the user requirements document, then to use knowledge of the domain setup, a technical document or specification, which is sent to the user for approval. Once this specification is returned this document is used as a base for development.

Converting the user requirements is a necessary task. “Specifications and contractual documents should be generated from the requirements” (Tavassoli, 2006, p5). This enhances the understanding of the requirements, while avoiding duplicated requirements and identifying requirements that were omitted.

Alternatively one can start development before the technical specification is approved, and then adjust the system afterwards if need be, as was the case at Company 3 and Company 2. Although developing a system without complete user requirements is possible and time is saved by not having to wait for user feedback, incomplete requirements present some challenges to the developers. Taking this into account it should thus be calculated how much time will be spent developing, and how much time will be spent afterwards on debugging the system.

Not only are the requirements necessary to develop the system, but the user requirements are also used at the end of the development process to determine whether the project is a success.

It might be difficult, but it is worthwhile to spend extra time on gathering all the user requirements. “Efforts in this direction can improve and accelerate return on investment” (Tavassoli, 2006, p9).

## **Proposition 2**

From the interviews it can be deduced that the gathering of user requirements is a necessary, but challenging step in the development process. The companies interviewed were aware that spending more time on requirement analysis would be beneficial to the development process, but this view was not always adhered to. This was mostly because of time constraints of the project, especially when waiting for the necessary user feedback. The companies that did not always do proper user requirements were the smaller companies. This was, as stated above, because of time constraints and the fact that the companies always tended to develop the same type of systems.

- **Design**

There was not much focus placed on system design, especially in the two smaller companies who did not generally have to develop large systems. The main elements that played a role during the design of a system were to identify and break up the system into modular parts, see how the system will fit within the existing networks and choose appropriate pre-developed code for reuse. Only one company modeled the system by using a flow diagram, this again being the biggest company of the three. The fact that the two smaller companies mostly developed the same type of systems and had a very stable development platform already setup, played a role in these companies not having to do much design.

## **Proposition 3**

Although formal design techniques were rarely used, some elements of system design did present themselves. It was necessary during design to determine how the new system would fit in with existing systems and networks. This as well as dividing the system into modular pieces allowed developers to decide on reusing pre-developed code.

- **Development**

Like any other development, systems in telecommunications go through certain phases. One gets the requirements, designs the system, builds it, tests it and deploys it. All of these phases have their own challenges. The process of developing the system code is not the most difficult part.

## **Modular Development**

All the systems that were developed were developed in modular parts. These were identified during design, then developed and tested. After this the modular pieces would be combined and tested again to see whether the system worked as a whole. Some of the systems, as in the case of Company 3, could consist of thousands of small, physically distributed applications. This modular nature of the systems allowed the companies to build up a proper development platform with pre-developed parts that could easily be reused in other systems.

## **Development platform and Reuse**

In the interview with company 3 and 2 it emerged that they use a development platform. This platform consists of a library of reusable applications that company 3 calls “The Engine”. When the client wants a new system developed, the ready made applications are chosen from the library, modified for the client's specific needs and used to build the new system. The platform has to be very stable. This means the applications in this library should be properly tested and qualified already.

Not only does reusing these applications reduce the development time, but it also simplifies the development process because it promotes modular development. Company 1 also reuses applications, but does not mention a set or library that is kept for this purpose. The reason for this is most probably because the systems developed by company 1 are more diverse, management type systems rather than the general structured systems as developed by Companies 2 and 3.

## **Development Languages**

A wide variety of programming languages are used in the telecommunications industry. At Companies 1 and 2 the Microsoft based dot.Net framework with VB was used, while Company 3 did not use any Microsoft products. The reason for this is that Company 3 develops lower level systems that communicate more with the network and only rarely require graphical user interfaces to interact with users.

The languages used by each company were:

Company 1: VB, Delphi, HTML

Company 2: VB

Company 3: Java, C, Delphi, XML, Object Pascal

In Company 3, Pascal was only used to develop non core applications. These non core applications are not essential to the functionality of the system.

#### **Proposition 4**

Modular development plays an important part when developing mobile telecommunications systems. Firstly it reduces the complexity of the systems being developed. Secondly it allows the company to build up a library of components to be reused when developing other systems; this is also called a development platform. Having a development platform is beneficial to the development process. It seems that only the two smaller companies made use of a development platform and this is most probably because of the type of software systems that these companies develop. Developing systems on this type of development platform are known as Software Product Lines. Several articles have been published (Matinlassi, 2004; Harsu, 2001; Weiss *et al.*, 1999; Bosh, 2000; Van Ommering & Bosch, 2002) about product-line methods and frameworks, but these will not be discussed in this study. The use of a particular development language depends on the type of system being developed. Both companies whose systems communicated with telecommunications users, made use of mainly Microsoft development languages, while the company who developed lower level, infrastructure systems, did not.

#### **• Testing**

The developed software must go through a stage of testing before it is released. Once the development is ready, it is loaded into a testing environment. This environment is safe and controllable. Here the system is tested and fixed before it is released.

One of the main reasons for testing is to ensure that the system communicates correctly to the host networks. Simulator tools are especially useful for this type of testing. This means that instead of having to wait for feedback from the other systems to say whether the product interacts correctly, the product can be tested on the simulator. Other tests include stress testing, to see if the system can handle big volumes. Also user acceptance testing is done to give the end users an opportunity to see exactly what the system does and whether it conforms to their requirements. After the system is released, it is monitored and tested constantly to ensure uptime. Short development times played a part in the testing of the systems as well. Sometimes systems were built as fast as possible and any errors were fixed afterwards.



### **Proposition 5**

All of the companies interviewed did product testing. For some, testing was a bigger issue than for others, but the general idea was that testing is a required activity for every system. The type of testing done was to ensure the system works, that it communicates correctly with other systems, and that it does what the user expects. In companies with a solid development platform, testing was sometimes neglected when development time became an issue.

- **Support and Maintenance**

Maintenance is done to keep the system up and running and thus keep the client happy. After deploying a system, Company 1 monitors the system and keeps a log of all errors received from the system users. Company 2 also has people who monitor the system after the system has been rolled out and in addition to errors logged by users, also do their own tests on the system. The developers solve critical errors immediately. While less important errors are addressed as soon as convenient. According to Company 3, the maintenance of telecommunication systems is critical and often projects are sold with a support and maintenance contract.

Maintaining a system does have some challenges. Company 1 states that it has the problem that maintenance takes some of the resources away from development (see Challenges). Site management and version controlling tools makes maintaining the system much easier.

### **Proposition 6**

Maintaining the system after release was another factor at all of the companies. Company 3 stated that system maintenance is “mission critical”. This is confirmed in April and Coallier (1995) who stated that defects in telecommunications systems are unacceptable.

#### **4.5.1.5 Techniques**

- **Specification Document**

A trend emerged that the one document that is always used in the development process is a general specification document. This document is created following user requirements analysis and then sent back to the user for approval. The specification document takes into account how the internal network of the business works and how the system will fit into that network. It also forms the basis from which the system is developed and thus serves as a means of communicating this information throughout the business.

One type of specification document which is used by Company 3, is the protocol specification, which describes the way the system should interact with other systems.

Elements in the specification include business rules, data flows and user requirements. It might also show why the users want these things. The user specification can later be used to create a flow diagram, as was the case at Company 1.

If the specification document is thought to be incomplete it is extended to a functional specification. This functional specification goes into more detail about what the system should do and its relationship to other systems.

### **Proposition 7**

The one document used by all the companies was a specification document, which was required to see exactly what the system is supposed to do and how it communicates with other systems. Company 1 also used a flow diagram and functional specification.

- **Tools Used**

Company 3 used some tools in their development for testing and statistics. It also referred to pre-built interfaces as tools. Most of the tools used were self developed. There are some “outside” tools used, mostly open source. These purchased tools are almost always “non-core”.

Simulator tools are used for testing purposes, while tools are also used for development purposes, such as code management and version controlling. Standard interface tools are used, which means that the same code can be used for different development by simply changing a few lines of code.

Whether Company 1 used tools depended on the project. The tools that they used were mostly project management tools, and these tools were only used for big projects.

Both Companies 1 and 2 made use of Microsoft products during development. Company 3, on the other hand, stated that once the workings of the application go deeper down into the network, one moves out of the MS tool base completely. This difference of opinion might be because of the fact that Company 3 develops lower level applications. “Ours is the software sitting inside the switch”.

### **Proposition 8**

All of the companies utilized tools in the development process. The tools used ranges from development tools, most of which have been developed in-house, to testing tools and project management tools.

The more user centered applications of Companies 1 and 2 were mainly built using Microsoft based tools while the network focused applications, especially at Company 3 were not Microsoft based.

### **4.5.2 Key aspects of mobile telecommunication software development**

#### **4.5.2.1 Flexibility and Adaptability**

According to Company 2, the development process must be kept flexible and adaptable. While Company 3 declared that they are very flexible in their approach, and that this approach is very advantageous. The cause of this extreme flexibility in Company 3 is the fact that they are a small company with only ten employees. This, combined with the fact that they use little documentation allows any changes to be addressed very quickly.

Part of Company 2's strategy for keeping a system flexible is to reuse ready developed code. Company 3 also focused on reuse, in this case having a whole library of pre-developed code that they could select to use in the system and modify where needed.

The need for system development to be flexible is because the companies are always working with other development teams, and the systems being developed are constantly communicating with other systems.

### **Proposition 9**

The main reason for keeping the development process flexible and adaptable was because of the interaction with development teams from other companies. Flexibility can be achieved by keeping the development team small, not producing too much documentation and looking to reuse ready developed code as much as possible.

#### **4.5.2.2 Development Time**

The time available to develop a telecommunications system is generally very short. Many times development times were said to be a few weeks. Because of these short development times there is not always time to draw up accurate specifications, or to do proper testing. Companies 1 and 2 believe that using formal systems development methodologies increased the time it takes to develop a system. In an industry where success is dependent on how fast one can roll out a product, the last thing one wants to do is get yourself stuck on formal procedures. Adherence to strict procedures and phases are time consuming and, depending on the project, not always necessary.

##### **Reason:**

One of the reasons why the time available for telecommunication development is so short is because of pressure from the main networks to deliver cutting edge products. Another reason, according to Company 1 is the rapid growth in technology.

##### **Decreasing development time:**

Being flexible and not having strict procedures could help bring down the time it takes to develop a system. Company 2 shared this view by saying that “big companies have lots of strict procedures, and the thing is... it takes forever” (C2). It should always be taken into account that there is a trade off between the speed at which one develops the system and the time takes to debug the system.

##### **Advantages of decreased development time:**

To keep customers happy, development time should be kept to a minimum.

##### **Obstacles:**

The time available for development can be used by many unforeseen circumstances such as developers having to stop what they are busy with and do maintenance on previously developed projects. The time spent on communicating with customers and other project teams caused much time to be wasted. This contributes to increased development time.

In the Telelogic white paper (Telelogic, 2006) they talk about projects called, “market-driven projects.” “Where getting a product to market on time is more important than delivering complete functionality”. It can be deduced from the above that time is a big factor in the development process and that telecommunication systems generally fall into this class of projects, as opposed to “contract-driven projects”. Where “conformance to requirements and compliance to regulations is

more critical.” (Telelogic, 2006, p3). Projects that fall into this second class usually have very long development times, because they have to adhere to strict policies.

#### **Proposition 10**

Time constraints on project development serve as a big challenge for the development companies. The lack of development time also seemed to be a big reason why the companies do not want to use formal systems development methodologies.

#### **4.5.2.3 Communication with other systems**

In any telecommunication system, one is constantly linked with other networks. This means that the system has to be synchronized between 2 or more systems all the time. An attribute of both Company 2 and 3's systems was that they kept them open and adaptable to be able to fit in with other networks. The problem with being dependent on other networks, as was the case at all three companies, is that sometimes these companies are much slower and less adaptable. This is a difficult problem to overcome. Because of the connection with other networks, it is important that one keeps track of the protocols used and to make sure one tests whether the system correctly interfaces with the other networks before launch. A new system's influence on the other systems should always be taken into account during design and development.

#### **Proposition 11**

Any system developed in the telecommunications industry will be in constant communication with other systems. This fact served as a big source of difficulty and frustration at Companies 2 and 3. These companies tried to keep their systems as flexible and adaptable as possible, so that they fit in easily with other systems.

#### **4.5.2.4 Environment**

- **The Telecommunications Industry**

The telecommunications industry is a challenging industry to work in. Company 1 stated that: “It is a very challenging environment” (C1). Company 2 stated that it is “one of the most difficult industries to work in.” (C2). “It is a very big, multi faceted environment” (C3).

Some characteristics of the telecommunications industry which were identified in the interviews, were:

- That the industry is controlled by many governing bodies, like ICASA
- In the telecoms industry, security is very important
- South African telecommunications is immature when compared to the rest of the world
- It is a good industry to be in because it is constantly growing
- The systems in telecommunications influence many more people than normal systems do
- Telecommunications is a “cut throat” market, where everything has to happen very quickly.

### **Proposition 12**

The telecommunications industry is a challenging industry, but also a good industry to work in. The reason for this is because, although it is a “cut throat” market where systems have to be developed very fast and the systems that are developed influence millions of people, telecommunications is constantly growing. Even though telecommunications in South Africa is still immature when compared to other countries, it has become an indispensable part of every day life.

- **Challenges**

Telecommunications is a challenging industry, with companies having to conform to the rules and regulations, as determined by the major networks and governing bodies. Companies 2 and 3 both mentioned the difficulties they experienced when communicating with outside networks. Company 3 stated that a possible solution to this problem was to apply commercial reasons for them to speed up communication. Not only do these systems interact with outside systems, but they also interact with systems within the company itself. “The number of features and their interactions are a major source of complexity in telecommunications software” (Mayrand, 1996, p60).

Some challenges identified by Company 1 were:

- The fact that developers were often taken off the development team to do maintenance on other systems
- Creating and maintaining a backup strategy was a challenging job
- Calculating risk, quality control, procurement and even legal aspects are all challenges that have to be faced in the telecoms industry

- Technology is constantly evolving (“Every 48 hours” (C1)) and you have to keep up to date with these changes
- Buying and the setup of hardware for the system to run on.

Company 2 stated that a big challenge was the fact that there are no strict, international standards concerning protocols that are to be used. The protocols that are used are either in-house, or propriety software.

Two more challenges mentioned by Company 3 were managing customer expectations (“The second biggest issue is customer expectations” (C3)) and gathering user requirements, which was said to be “Painful” (C3). Company 1 concurred with Company 3 on the subject of user requirements. This might be because neither of them have a strict user requirements gathering phase and they end up with sub standard user requirements. It is difficult to develop a system from that.

The time that is available to develop a system is constrained. Many times development times were said to be a few weeks.

### **Proposition 13**

Two of the big challenges that emerged from the data were the communication between different development teams and the short development times. The gathering of user requirements was also considered to be a challenge. Other challenges were, system complexity, the assignment of resources managing customer expectations and the lack of proper standards and many more.

#### **4.5.3 Identify a list of necessities a systems development methodology should contain when used to develop mobile telecommunications software**

Here the researcher will discuss whether the companies adhered to the list of software development needs, which was presented in Chapter 2, during their development process. Any thoughts the interviewees had about the items discussed will also be mentioned. Any additional elements that were identified from the interviews and should be addressed by the systems development methodology will be discussed at the end of the section.

#### **4.5.3.1 Adherence to list of software development needs**

- **Well-documented analysis and design phase**

There was very little emphasis placed on a specific analysis and design phase during product development. All of the companies did, however, analyse the specification document, which was generated after the user requirements phase to see how the new systems would fit into the current systems and networks of the company, after which the systems were also divided into modular parts.

A well-documented analysis and design phase is more important in a bigger company that produces bigger, management type systems, as is the case with Company 1. Company 1 also partly documented this phase by means of flow diagrams.

- **Prototyping**

No mention was made by any of the interviewees regarding prototyping.

- **Automatically generate code, prototypes and specifications**

No mention was made by any of the interviewees regarding the automatic generation of code, prototypes or specifications.

- **Formal document notation**

Only the general specification document was used by all three of the companies, while only the biggest of the three companies used flow diagrams. No formal document notation was used, however, and Company 3 specifically stated that they used as little documentation as possible in order to keep their development process flexible.

- **Strenuous testing and review sessions**

All of the companies interviewed did product testing. For some, testing was a bigger issue than for others, but the general idea was that testing is a required activity for every system. The type of testing done was to ensure the system works, that it communicates correctly with other systems, and that it does what the user expects.

- **Reuse**

Reuse of both code and components were important factors during the development process of all the companies. Companies 2 and 3 used and maintained a library of pre-built



applications and interfaces that they reused during the development of other systems. Reusing pre existing code also helped Company 2 to keep their systems flexible and scalable.

- **Distribution**

The issue of distribution was mentioned only by Company 3, which was the only company that developed low level systems. They stated that some of the systems could consist of thousands of small programmes, on different machines, that have to work together. The company handled this distribution by reusing ready made and tested pieces of code to handle the interfaces. These pre-developed interfaces make up a part of the development platform used at Company 3.

- **Scalable product**

Companies 1 and 3 named the growth of the telecommunications industry as an important challenge. The interviewees said that the industry is constantly evolving and that new products and services need to be added to the systems already running. Company 2 also mentioned the industry growth and thought it to be a good reason to be working in telecommunications. The issue of product scalability was not discussed in any of the interviews, but once again it can be seen that by using a development platform, as at Companies 2 and 3, any additional functionality can just be “slotted” into place on the existing system.

- **Flexible and Adaptable product**

Flexibility and adaptability were discussed for the product being developed as well as the development process. The product has to be flexible and adaptable because the system is in constant contact with other networks and systems, which could change with short notice. This means the system has to be able to handle any changes in the systems it is in communication with. The systems development process has to be adaptable because one has to work with development teams from other companies that sometimes take longer to get things done. Added to this is the fact that the systems development methodology has to take into account any unforeseen circumstances that might arise during development. Once again the use of a development platform was given credit for keeping the product flexible and adaptable. Company 3 attributed their flexibility during the development process to the small number of employees and their sparse use of documentation.

- **Reduce cost of project**

Some of the biggest issues effecting project cost were mentioned by Company 1 as being the setup of proper security measures and keeping the system in working order. Having proper security measures added to the systems and proper maintenance once the system is developed should help in reducing the overall cost. There are, however, many elements that could contribute to decreasing the development cost, not the least of which is shorting the development time and reusing pre-developed code.

- **Communication medium**

The general specification document was used as the main communication medium in all of the companies.

- **Reduce complexity of product**

In order to reduce the complexity of the system being developed, all the companies broke up the system into more manageable pieces. These pieces were then modularly developed and tested, after which all the pieces were combined. Company 1 added that using a complex systems development methodology only increased the complexity of the entire development process and they thus tried to use as simple a systems development methodology as possible.

- **Reduce development time**

Short development time was considered by all the interviewees to be a major issue during the development of mobile telecommunication software, if not the biggest issue. The companies all described possible ways of reducing the overall development time. Being flexible is a way to reduce development time according to Company 3. The development time was made even shorter by not waiting for user requirements to be approved before starting development and using very little documentation. Company 1 reduced development time by keeping the development process simple and uncomplicated. Being too strict and using formal systems development methodologies were factors that increased the development time according to Companies 1, 2 and 3.

#### **4.5.3.2 Additional software development needs identified from the interviews**

- **Constant communication with other systems**

Telecommunication systems are in constant contact and are constantly influenced by other networks and systems. Likewise the systems themselves also impose an influence on other systems. This should be taken into account when designing and developing any new system. The constant communication with other systems is also a good reason for keeping the system being developed as flexible and adaptable as possible. All of the companies took the influence the new system would have on any current systems into account during design and development. When working with other systems it is important to keep track of any protocols being used. For this purpose Company 3 used a protocol specification document.

- **Modular development**

Although modular development was used as a way of reducing the complexity of the system, the researcher believed it to be important enough to be highlighted as an issue on its own. This is because all of the companies interviewed first broke the system down into modular pieces before starting development.

- **Maintenance and support**

The interviewees at Company 3 noted that one of the elements that set mobile telecommunication systems apart from systems in other industries is the number of people influenced if the system fails. Mobile telecommunication systems could influence thousands if not millions of users. In order to ensure the system operates correctly, maintenance should be done on all developed systems. All of the companies interviewed did extensive maintenance and believed it to be a very important aspect of mobile telecommunication system development. Company 3 always sells support with each system and monitors system functionality from one centralized location using statistics tools. Companies 1 and 2 continually test each system after release to ensure the system functions and to fix any problem as soon as possible.

#### 4.5.3.3 Summary

After reviewing the list of development needs and identifying whether they were deemed important as well as how they were addressed within each company, the following can be deduced. Although some of the issues on the list were deemed important, others were not mentioned at all during the interviews. The three most important issues were identified as being flexibility and adaptability, reuse of pre-developed code and components, and decreasing the development time. Two issues that were not so important were prototyping and the automatic generation of code and specifications. Three additional issues were identified from the interviews and added to the list. To answer the original research question of “Identify a list of necessities a systems development methodology should contain when used to develop mobile telecommunications software,” a list of development needs is given below:

- *Well-documented analysis and design phase*
- Strenuous testing and review sessions
- Reuse
- Distribution
- Scalable product
- Flexible and Adaptable product
- Reduce cost of project
- Communication medium
- Reduce complexity of product
- Reduce development time
- Constant communication with other systems
- Modular development
- Maintenance and support.

The first item on the list is printed in italics because, although analysis and design was an issue at all of the companies, very little emphasis was placed on this phase.

This list was then sent to all the interviewees and they were asked to say whether they agreed with the issues that were identified. They all agreed that the issues were relevant and that they tried to address all of these issues while developing a new system.

## **4.6 Conclusion**

In this chapter the findings of the case study as well as the proposition deduced by means of the cross-case analysis method were given. This was done in order to determine how mobile telecommunications systems are developed in practice. The researcher specifically studied the use of systems development methodologies during the development process, key elements of telecommunication software development and also reviewed and updated the list of software development needs discussed at the end of Chapter 2.

In the following chapter the results and final conclusions of the study will be given.

## **CHAPTER 5**

### **SUMMARY AND FINAL CONCLUSIONS**

#### **5.1 Introduction**

The purpose of this study was to determine how mobile telecommunication systems are being developed in South Africa. The main focus was on the use of possible system development methodologies during the development process. This was done by first conducting a literature study of the current methodologies available, as well as some of the methodologies that are currently being used within the telecommunications domain. After this, case studies were done at three different mobile telecommunications development companies to determine how mobile telecommunications systems are being developed in practice. In this chapter results obtained from this study will be reviewed briefly.

#### **5.2 Research contributions**

The mobile telecommunications industry is one of the fastest growing industries in South Africa. While this may be the case, it was found that there is very little available literature on the subject of mobile telecommunications development methodologies. This study aims to identify whether an existing formal methodology could be used to develop computer systems in the mobile telecommunications environment successfully.

This study looked at the following issues:

- Firstly a literature study on system development methodologies was done. System development methodologies were defined and several methodologies that are currently available for use were discussed
- Secondly three methodologies that are used in practice were described. These methodologies are: The MODA-TEL and Mobile-D methodologies, as well as a methodology that utilizes formal methods to accelerate development of telecommunication software, as described in Mansurov (2000)
- Thirdly these methodologies were compared to determine the similarities and differences between the methodologies, as well as to better understand how and why they are used

- Fourthly some of the key characteristics of the mobile telecommunication environment were viewed and a list of necessary elements which should be addressed in a mobile telecommunications systems development methodology was compiled
- Fifthly a description of the research approach, methods and techniques was given while doing the field research for this study. Here it came to light that the qualitative case study method was used, utilizing interviews to retrieve data from various telecommunication companies. The data gathered from the interviews were then coded and grouped in the computer software programme ATLAS.it and analyzed using the cross-case analysis method
- Finally the analyzed interview data as well as the findings after analysis were presented. The cross-case matrix used during the cross-case analysis process is shown in Appendix A.

There are many definitions for a systems development methodology and in this study a systems development methodology is defined as a combination of:

- Approach(es)  
This is the philosophy on which the methodology is built. Different approaches could be concerned with cost, quality documentation, development speed, adaptability or efficiency. Examples of system development approaches are the structured approach, object-oriented approach, information modeling
- Process model(s)  
Every model defines a certain order or sequence of stages through which a system evolves (Wynekoop & Russo, 1993). Examples are the linear lifecycle model and the Boehm's spiral model
- Method(s)  
A method is a systematic approach to conducting at least one complete phase of system development, consisting of a set of guidelines, activities, techniques and tools, based on a particular philosophy of systems development and the target system (Wynekoop & Russo, 1993). Examples are OMT, IE

- Technique(s)

The procedure used to perform a development activity (Brinkkemper, 1996). An example of a technique is the entity relationship diagram.

### **5.3 Research aims and objectives**

The aims of the study were to determine how mobile telecommunication systems are developed in South Africa by focusing on the use of systems development methodologies. In order to determine this the following three main research objectives were stated:

- To identify the systems development methodologies used, if any, by companies in South Africa in the development of mobile telecommunication systems
- To highlight certain key aspects of mobile telecommunication software development and
- To identify a list of necessities a systems development methodology should contain when used to develop mobile telecommunications software

### **5.4 Results of the study**

The results below will be discussed in the order of the list of objectives stated above. Firstly the propositions regarding the systems development methodologies used by companies in South Africa will be given. Secondly, the propositions regarding key aspects of mobile telecommunication software development will be discussed. Thirdly, results of the list of mobile telecommunications software development methodology necessities will be discussed.

#### **5.4.1 Systems development methodology used**

It was found that mobile telecommunications software development companies in South Africa tended to make use of in-house developed systems development methodologies, focusing on flexibility and speed of development. This was driven by the constant interaction of software with other telecommunications systems and the speed at which mobile telecommunication software need to be developed. This focus on flexibility and speed was similar to the mobile telecommunications systems development methodologies discussed in Chapter 2. Other similarities include the modular development style and the phases that were followed during the development process. Modeling of the system, however, did not play such an important role in the in-house developed systems development methodologies



- **Proposition 1**

In this study it was found that telecommunication systems development companies tend to use their own, in-house systems development methodology, because the formal systems development methodologies were believed to be too theoretical and only beneficial to big companies. These in-house systems development methodologies were specifically designed to fit in with the character of the company and more than that, were able to change depending on the project. By using some form of a systems development methodology, more control could be gained over the entire project life cycle. Another advantage is the fact that it could decrease the time it takes to develop a system. Sticking to a formal, rigid, systems development methodology though, was thought to have a negative impact on development time.

- **Proposition 2**

The gathering of user requirements is a necessary but challenging step in the development process. Spending more time on requirement analysis would be beneficial to the development process, but this view was not always adhered to. This was mostly because of time constraints of the project, especially when waiting for the necessary user feedback. The companies that did not always do proper user requirements were the smaller companies. This was, as stated above, because of time constraints and the fact that the companies always tended to develop the same type of systems.

- **Proposition 3**

Although formal design techniques were rarely used, some elements of system design did present themselves. It was necessary during design to determine how the new system would fit in with existing systems and networks. This as well as dividing the system into modular pieces, allowed developers to decide on reusing pre-developed code.

- **Proposition 4**

Modular development plays an important part when developing mobile telecommunications systems. Firstly it helps to reduce the complexity of the systems being developed. Secondly it allows the company to build up a library of components to be reused when developing other systems; this is also called a development platform. Having a development platform is beneficial to the development process. The research shows that only the two smaller

companies made use of a development platform, this is most probably because of the type of software systems that these companies develop. Developing systems on this type of development platform is known as “Software Product-Lines.” Several articles have been published (Matinlassi, 2004; Harsu, 2001; Weiss *et al.*, 1999; Bosh, 2000; Van Ommering & Bosch, 2002) about product-line methods and frameworks, but these will not be discussed in this study. The use of a particular development language depends on the type of system being developed. Companies whose systems were located in the “Telecommunications Users” or “Service Provider” layers of the service model discussed in chapter 1 made use of mainly Microsoft development languages, while the company who developed lower level systems located in the “Telecommunications Infrastructure” layer, did not.

- **Proposition 5**

Testing is a required activity for every system that is developed. The type of testing done was to ensure that the system works, that it communicates correctly with other systems, and that it does what the user expects. In companies with a solid development platform, testing was sometimes neglected when development time became an issue.

- **Proposition 6**

The maintenance of system after release is a major factor in the telecommunications industry and can be considered “mission critical”, while any errors or defects are unacceptable. Part of this lies in the fact that the number of people effected if a telecommunication systems fails could be thousands or even millions.

- **Proposition 7**

Documentation during the development process was not a high priority in mobile telecommunication systems development. Documents were used more at bigger companies who tended to develop larger systems. Some of the documents used included flow diagrams, specification documents, functional specs and protocol specs.

- **Proposition 8**

Tools were used during development, although not extensively. The tools used ranged from development tools, most of which had been developed in-house, to testing tools and project management tools. More user centered applications were built using Microsoft based tools

while tools used during the development of infrastructure focused applications were not Microsoft based.

#### **5.4.2 Key aspects of mobile telecommunication software development**

- **Proposition 9**

Flexibility and adaptability play a very important part in mobile telecommunication systems development. This is not only from a product perspective, where the system being developed should be able to communicate with other systems and adapt to any changes, but also from a development point of view. This is because of the interaction with development teams from other companies, which is a necessity while developing a system. Flexibility is achieved by keeping the development team small, not producing too much documentation and looking to reuse ready developed code as much as possible.

- **Proposition 10**

One of the single biggest issues during the development of mobile telecommunication systems was the short amount of time available for development. This was attributed to the competitive nature of the telecommunications industry. It affects all companies, big and small, regardless of the type of systems being developed. The time available for development has a direct effect on how much effort is put into the gathering of user requirements, design and testing. It is also a big reason why formal systems development methodologies are not used.

- **Proposition 11**

Any system developed in the telecommunications industry will be in constant communication with other systems. It is also a fact that any systems being developed will have to adapt and conform to the main networks into which they are introduced. Companies should try to keep their systems as flexible and adaptable as possible, so that they fit in easily with other systems.

- **Proposition 12**

The telecommunications industry is a challenging industry, but also a very good industry to work in. The reason for this is because, although it is a “cut throat” market where systems have to be developed very fast and the systems that are developed influence millions of

people, telecommunications are constantly growing. Even though telecommunications in South Africa are still immature when compared to other countries, it has become an indispensable part of every day life.

- **Proposition 13**

Two of the big challenges companies face when developing a mobile telecommunications system are the issues of communication between different development teams and the short development times. The gathering of user requirements can also be considered as quite a challenge. Other challenges include system complexity, the assignment of resources, managing customer expectations, the lack of proper standards and many more. Except for the challenges faced during development, maintaining live systems is made more difficult by the fact that the industry is constantly growing and new technology and services are being introduced almost on a daily basis.

#### **5.4.3 A list of necessities a systems development methodology should contain when used to develop mobile telecommunications software**

The thirteen items in the list below were identified as being the top necessities a systems development methodology should contain when used to develop a mobile telecommunications system:

- **Analysis and design phase**

The analysis and design phase should be present to ensure the system integrates with all existing networks and systems. Breaking up the system into modular parts is also an important aspect of the design phase

- **Strenuous testing and review sessions**

The systems development methodology should allow for every system to be tested before going live, and tested even more rigorously after the system has been released

- **Reuse**

The reuse of pre-developed code plays an important part in decreasing the time it takes to develop a system. The systems development methodology used should facilitate building up a library of developed code and development components, while also identifying when these components could be reused

- **Distribution**

One inherent feature of the mobile telecommunications environment is distribution. Techniques and procedures should be available to identify and address the issue of system distribution

- **Scalable product**

The rapid growth of the industry is a major challenge and the issue of scalability should be addressed by the systems development methodology used during the development process

- **Flexible and Adaptable product**

The systems development methodology used should be flexible and adaptable because development needs may change from one project to the next. In addition the systems development methodology should aim to develop systems which are flexible and adaptable. This is because of a multitude of factors from the systems dependence on other systems to the introduction of new services and technology

- **Reduce cost of project**

Reducing the cost of systems development is a big business concern in any industry and the reduction of project cost should be a major focus of any systems development methodology

- **Communication medium**

It is necessary for a company to have some way of communicating the system details and requirements to all the developers. This is usually done by means of documentation

- **Reduce complexity of product**

Telecommunications systems are inherently complex, with many factors adding to the level of complexity. A good systems development methodology should help to reduce the complexity of the system being developed

- **Reduce development time**

Because of the constraint on available development time in the mobile telecommunications industry, any systems development methodology should help to increase the speed of developing the system

- **Constant communication with other systems**

The systems developed in the mobile telecommunications industry have the characteristic that they are in constant communication with other systems and networks. This should all be taken into account when designing and developing the system

- **Modular development**

Any systems development methodology used during mobile telecommunications systems development should simplify and facilitate modular development. Modular development is considered such an important element because it reduces complexity, helps keep systems flexible and scalable, while also promoting reuse

- **Maintenance and support**

Maintaining systems that have been developed is extremely important in the telecommunications industry and should be a part of the systems development methodology used.

## **5.5 Limitations of the study and future work**

One of the main limitations of this study was the number of cases studied. Three cases are not enough to build a proper theory on the use of systems development methodologies in the mobile telecommunication industry. Future work should address this limitation by using more cases. Another limitation was the fact that there were too few interviewees. The researcher minimized the negative effect of this limitation by performing in-depth interviews as well as studying any documentation relevant to the companies or their systems development process.

Together with the above, a big limitation was the focus of the study concerning types of systems being developed (See the discussion of the service model in Chapter 1). This study aimed to study a generalised systems development methodology used to develop all the different types of systems developed in the telecommunications industry. It became clear from the research that different types of systems would require different systems development methodologies. This could be addressed in future research by focusing on only one type of system, or the systems developed for a specific layer of the telecommunications environment (as shown in Chapter 1).

Future work should aim at extending the list of software development needs and environmental elements, by adding more elements as well as determining the importance of each element in the

list. This is a purely exploratory study and future work should be more in-depth and of a quantitative nature.

## BIBLIOGRAPHY

- ABBAS, R., KAZMIERCZAK, F. & DART, P. 1998. Software architecture for collaborative development: a mobile phone network simulation example. (Paper delivered as a report no. 14, for Department of Computer Science, The University of Melbourne on September 1998. 19p.)
- ABRAHAMSSON, P., HANHINEVA, A., HUIKKO, H., IHME, T., JÄÄLINOJA, J. & KORKALA, M. 1994. Mobile-D: an agile approach for mobile application development. (In 19th Annual ACM Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA'04). Vancouver:ACM. p.174-175.)
- ABRAHAMSSON, P., SALO, O., RONKAINEN, J. & WARSTA, J. 2002. Agile software development methods. Review and analysis. *VTT publications*, 478p. <http://www2.umassd.edu/swpi/xp/papers/p478.pdf> Date of access: 21 Feb. 2006.
- APRIL, A. & COALLIER, F. 1995. TRILLIUM v3.0: a model for the assessment of telecom software systems development capability. (In Rout, T., ed. Proceedings 2<sup>nd</sup> international SPICE symposium. June 1-2 1995. Brisbane. p.79-88)
- AVISON, D. & FITZGERALD, G. 2002. Information system development: methodologies, techniques and tools. Berkshire: McGraw-Hill. 416p.
- BARRY, C. 1998. Choosing qualitative data analysis software: Atlas/ti and Nudist compared. *Sociological research online*, 3. <http://www.socresonline.org.uk/3/3/4.html> Date of access: 13 Jan. 2006.
- BOSCH, J. 2000. Design and use of software architectures: adopting and evolving a product-line approach. Harlow: Addison-Wesley. 368p.
- BRINKKEMPER, S. 1996. Method engineering: engineering of information systems development methods and tools. *Information and software technology*, 38: 275-280.
- BRYMAN, A. 2004. Qualitative research on leadership: a critical but appreciative review. *The leadership quarterly*, 15: 721-891.



- BURNS, R. 2000. Introduction to research methods. London: Sage. 624p.
- CHECKLAND, P. & SCHOLLES, J. 1999. Soft systems methodology in action. Chichester: Wiley. 418p.
- CHUA, W. 1986. Radical developments in accounting thought. *The accounting review*, 61: 601-632.
- COAD, P. & YOURDON, E. 1991. Object oriented analysis. 2nd ed. Englewood Cliffs, N.J.: Prentice Hall. 233p.
- COALLIER, F., GRAYDON, A. & FICCICI, M. 1995. Trillium: model for telecom product development & support process capability, Release 3.0. Bell Canada. 118p. <http://ricis.cl.uh.edu/trillium> Date of access: 3 Oct. 2006.
- CRESWELL, J. 1997. Qualitative inquiry and research design: choosing among five traditions. Thousand Oaks: Sage. 424p.
- DARKE, P., SHANKS, G. & BROADBENT, M. 1998. Successfully completing case study research: combining rigour, relevance and pragmatism. *Information systems journal*, 8: 273-290.
- DE VRIES, M. 2004. Evaluating or selecting a suitable information systems development methodology: a case study. *SA journal of industrial engineering*, 15: 9-25.
- DSDM (Dynamic systems development method). 2003. DSDM public version 4.2 manual. DSDM Consortium. <http://www.dsdm.org/version4/2/public> Date of access: 29 Aug. 2005.
- EISENHARDT, K. 1989. Building theories from case study research. *Academy of management review*, 14: 532-550.
- FERNÁNDEZ, W. D. 2004. Using the glaserian approach in grounded studies of emerging business practices. *Electronic journal of business research methods*, 2: 83-94.
- FITZGERALD, B. 1996. Formalized systems development methodologies: a critical perspective. *Information systems journal*, 6: 3-23.

GANE, C. & SARSON, T. 1979. Structured systems analysis: tools and techniques. Englewood Cliffs, N.J.: Prentice Hall. 288p.

GAVRAS, A., BELAUNDE, M., PIRES, L. & ALMEIDA, J. 2004. Towards an MDA-based development methodology. (*In* van Sinderen M.J & Ferreira, L., *ed.* 1st European workshop on model-driven architecture with emphasis on industrial applications. Centre for Telematics and Information Technology, University of Twente. Mar. 2004. p.71-81)

GERVAIS, M. & RUFFEL, N. 1997. Design of telecommunication services based on software agent technology and formal methods. *GLOBECOM*, 3: 1724-1728.

GILGUN, J. F. 1992. Definitions, methodologies, and methods. (*In* Gilgun, J., Daly, K. & Handel, G., *ed.* Qualitative family research. Thousand Oaks: Sage. p. 22-29.)

GLASER, B. G. 1998. Doing grounded theory: issues and discussions. Mill Valley: Sociology Press. 254 p.

GLASSER, P. 1998. The knowledge factor. *Chief information officer*, 12: 108-118.

GOLUBIC, S. & MARRUSIC, D. 1999. Reviews and inspections: an approach to the improvement of telecom software development process. [www.cas.mcmaster.ca/wise/wise01/GolubicMarusic.pdf](http://www.cas.mcmaster.ca/wise/wise01/GolubicMarusic.pdf)  
Date of access: 27 Jun. 2006.

HALONEN, T., ROMERO, J. & MELERO, J. 2003. GSM, GPRS and EDGE performance, evolution towards 3G/UMTS. West Sussex: Wiley. 654p.

HANCOCK, B. 1998. Trent Focus for research and development in primary health care: an introduction to qualitative research  
[http://faculty.uccb.ns.ca/pmacintyre/course\\_pages/MBA603/MBA603\\_files/IntroQualitativeResearch.pdf](http://faculty.uccb.ns.ca/pmacintyre/course_pages/MBA603/MBA603_files/IntroQualitativeResearch.pdf) Date of access: 3 Nov. 2005.

HARSU, M. 2001. A survey of product-line architectures, report 23. Tampere University of Technology: Institute of Software Systems. 79p. <http://practise2.cs.tut.fi/pub/papers/arkki.pdf> Date of access: 26 Jun. 2005.

- HOEPFL, M. C. 1997. Choosing qualitative research: a primer for technology education researchers. *Journal of technology education*, 9: 47-63.
- HOLLER, R. 2006. Mobile application development: a natural fit with agile methodologies. Alpharette, GA:VersionOne. <http://www.versionone.net/pdf/MobileDevelopment.pdf> Date of access: 22 Jan. 2007.
- HOWARD, A. 1997. A new RAD-based approach to commercial information systems development: the dynamic systems development method. *Industrial management & data systems*, 97: 175-177.
- HUISMAN, M. & IIVARI, J. 2006. Deployment of systems development methodologies: perceptual congruence between IS managers and systems developers. *Information & management*, 43: 29-49.
- HUSSEY, J. & HUSSEY, R. 1997. Business research: a practical guide for undergraduate and postgraduate students. London: Macmillan Press. 368p.
- IIVARI, J., HIRSCHHEIM, R. & KLEIN, H. 2000. A dynamic framework for classifying information systems development methodologies and approaches. *Journal of management information systems*, 17: 179-218.
- JACOBSON, I., CHRISTERSON, M., JONSSON, P. & OVERGAARD, G. 1992. Object-oriented software engineering: a use case driven approach. Workingham: Addison-Wesley. 552p.
- JEFFRIES, R. 2001. Extreme programming installed. Harlow: Pearson Education. 288p.
- KALAYANASUNDARAM, S., PONNAMBALAM, K., SINGH, A., STACEY, B. J. & MUNIKOTI, R. 1998. Metrics for software architecture: a case study in the telecommunication domain. *Proceedings of the IEEE Canadian conference on electrical and computer engineering*, 2: 715-718.
- KLEIN, H. & MYERS, M. 1999. A set of principles for conducting and evaluating interpretive field studies in information systems. *MIS quarterly*, 23: 67-93.
- KNUTSON, J. 2003. Whether to waterfall or wave. *PM network*, 17: 74-76.

- KOHN, L. T. 1997. Methods in case study analysis. Technical publication number 2. The Center for Studying Health System Change. <http://www.hschange.org/CONTENT/158/158.pdf> Date of access: 5 May 2005.
- KRUCHTEN, P. 2000. The rational unified process an introduction. 2nd ed. Reading, MA: Addison-Wesley. 320p.
- LAL, D., PITT, D. & BELOUCIF, A. 2001. Restructuring in European telecommunications: modeling the evolving market. *European business review*, 13: 152-158.
- LAURIDO-SANTOS, O. 1986. A course on information engineering. *ACM SIGCSE bulletin*, 18: 194-202.
- MADDISON, R., BAKER, G.J., BHABUTA, L., FITZGERALD, G., HINDLE, K., SONG, J.H.T., STOKES, N. & WOOD, J.R.G. 1983. Information system methodologies. Chichester: Wiley Heyden. 415p.
- MAHONEY, C. 1997. Common qualitative methods. (In Fretchling, J. & Sharp, L., ed. User-friendly handbook for mixed method evaluations. National science foundation. Part II. Chapter 3. [http://www.ehr.nsf.gov/EHR/REC/pubs/NSF97-153/pdf/mm\\_eval.pdf](http://www.ehr.nsf.gov/EHR/REC/pubs/NSF97-153/pdf/mm_eval.pdf) Date of access: 28 May 2005)
- MANSUROV, N. 2000. Formal methods for accelerated development of telecommunications software. <http://www.ispras.ru/groups/case/downloads/methodology.pdf> Date of access: 29 Mar. 2006.
- MARTIN, J. 1989. Information engineering. Englewood Cliffs, N.J: Prentice Hall. 268p.
- MARTIN, J. 1991. Rapid application development. Englewood Cliffs, N.J: Prentice Hall. 736p.
- MARTIN, P. Y. & TURNER, B. A. 1986. Grounded theory and organizational research. *The journal of applied behavioral science*, 22: 141-157.
- MATHERS, N., FOX, N. & HUNN, A. 1998. Using interviews in a research project [http://www.trentfocus.org.uk/Resources/Using Interviews.pdf](http://www.trentfocus.org.uk/Resources/Using%20Interviews.pdf) Date of access: 3 Jun. 2005.

MATINLASSI, M. 2004. Comparison of software product line architecture design methods: COPA, FAST, FORM, Kobra and QADA. (In 26th international conference on software engineering (ICSE'04). Los Alamitos CA:IEEE Computer Society. p.127-136.)

MAYRAND, J. 1996. Assessment of tools in the telecommunications industry: a customer perspective. (In Proceedings of SAST '96. Quebec:IEEE. p.69-70.)  
<http://csdl.computer.org/comp/proceedings/sast/1996/7389/00/73890069.pdf> Date of access: 15 Jun. 2005.

MEHRA, V. 2005. Achieving operational excellence in mobile telecom. <http://www.syntelinc.com>  
Date of access: 19 Apr., 2006.

MIDDLETON, P. 2000. Barriers to the efficient and effective use of information technology. *The international journal of public sector management*, 13: 85-99.

STEINHAU, R., ed. 2003. Deliverable d3.1: model-driven architecture definition and methodology. <http://www.modatel.org/public/deliverables/D3.1.htm> Date of access: 28 Mar. 2006.

MUMFORD, E. 1995. Effective requirements analysis and systems design: the ETHICS method. Basingstoke: Macmillan. 263p.

MYERS, M. 2003. Qualitative research in information systems. <http://www.qual.auckland.ac.nz/index.htm> Date of access: 7 Nov. 2006.

NEUMAN, W. 2003. Social research methods: qualitative and quantitative approaches. 5th ed. Sydney: Allyn & Bacon. 592p.

NEVILLE, C. 2005. Introduction to research and research methods. <http://www.brad.ac.uk/acad/management/external/els/pdf/introductiontoresearch.pdf> Date of access: 30 Apr. 2006.

PATEL, A. 2002. Current status and future directions of software architectures for telecommunications. *Computer communications*, 25: 121-132.

- PURVIS, R. & SAMBAMURTHY, V. 1997. An examination of designer and user perceptions of JAD and the traditional IS design methodolog. *Information and management*, 32: 123-135.
- ROBERTS, T., GIBSON, M., FIELDS, K. & RAINER, K. 1998. Factors that impact implementing a system development methodology. *IEEE transactions on software engineering*, 24: 640-649.
- RUBIN, H. & RUBIN, I. 1995. Qualitative interviewing: the art of hearing data. Thousand Oaks, CA: Sage. 304p.
- RUEGG, R. 2006. Bridging from project case study to portfolio analysis in a public r&d program: a framework for evaluation and introduction to a composite performance rating system. National institute of standards and technology. <http://www.atp.nist.gov/eao/gcr06-891/gcr06-891report.pdf>  
Date of access: 10 Nov. 2006.
- SANDERS, G., THORENS, L., REISKY, M. R. & O. DEYLITZ, S. 2003. GPRS networks. Chichester: Wiley. 304p.
- SEAMAN, C. B. 1999. Qualitative methods in empirical studies of software engineering. *IEEE transactions on software engineering*, 25: 557-572.
- SHARP, L. & FRECHTLING, J. 1997. Introduction to mixed method evaluations. (In Fretchling, J. & Sharp, L., *ed.* User-friendly handbook for mixed method evaluations. National Science Foundation. Part I, Chapter 1. [http://www.ehr.nsf.gov/EHR/REC/pubs/NSF97-153/pdf/mm\\_eval.pdf](http://www.ehr.nsf.gov/EHR/REC/pubs/NSF97-153/pdf/mm_eval.pdf)  
Date of access: 28 May 2005.)
- SOY, S. K. 1996. The case study as a research method.  
<http://www.gslis.utexas.edu/~ssoy/usesusers/l39ld1b.htm> Date of access: 8 Sep. 2006.
- STRUASS, A. & CORBIN, J. 1990. Basics of qualitative research: grounded theory procedures and techniques. Thousand Oaks, CA:Sage. 272p.
- TAVASSOLI, D. 2006. Ten best practices for writing better requirements. <http://www.telelogic.com>  
Date of access: 10 Nov. 2006.

- TELELOGIC. 2006. Requirements-driven development. <http://www.telelogic.com> Date of access: 10 Nov. 2006.
- THOMAS, R. & BRUBAKER, D. 2000. Theses and dissertations: a guide to planning, research, and writing. Greenwood: Gergin Garvey. 312p.
- TROCHIM, W. M. 2006. Qualitative approaches. Research methods knowledge base. <http://www.socialresearchmethods.net/kb/qualapp.htm> Date of access: 2 Feb. 2007.
- VALLECILLO, A. 2001. RM-ODP: the ISO reference model for open distributed processing. *DINTEL edition on software engineering*, 3: 69-99.
- VAN OMMERING, R. & BOSCH, J. 2002. Widening the scope of software product lines: from variation to composition. (In Gary, C., ed. The proceedings of the second product line conference, SPLC2. San Diego, CA. p.328-347.)
- VOSS, C., TSIKRIKTSIS, N. & FROHLICH, M. 2002. Case research in operations management. *International journal of operations & production management*, 22: 195-219.
- VTT. 2006. Agile software technologies. <http://virtual.vtt.fi/virtual/agile/mobiled.html> Date of access: 9 Sep., 2006.
- WALSHAM, G. 1993. Interpreting information systems in organisations. Chichester: Wiley. 269p.
- WALSHAM, G. 1995. Interpretive case studies in IS research: nature and method. *European journal of information systems*, 4: 74-81.
- WEAVER, P., LAMBROU, N. & WALKLEY, N. 1998. Practical SSADM version 4+: a complete tutorial guide. 2nd ed. London: Pitman. 403p.
- WEISS, D., LAI, C. & TAU, R. 1999. Software product-line engineering: a family-based software development process. Reading, MA: Addison-Wesley. 384p.
- WEST, D. Rational Software. 2002. Planning a project with the rational unified process. 11p. (White paper TP151, 08/02) [http:// www.imsi-pm.com/home/library/Planning\\_an\\_IT\\_Project\\_with\\_RUP.pdf](http://www.imsi-pm.com/home/library/Planning_an_IT_Project_with_RUP.pdf) Date of access: 14 Aug. 2005

WYNEKOOP, J. & RUSSO, N. 1993. System development methodologies: unanswered questions and the research practice gap. (In Proceedings of the 14th international conference on information systems. New York: ACM. p.181-190.)

YIN, R. K. 2003. Case study research, design and methods. Thousand Oaks: Sage. 131p.

YOURDON INC. 1993. Yourdon systems method: model-driven systems development. Englewood Cliffs, N.J: Yourdon Press. 682p.



## Appendix A

### Cross case analysis matrix

This matrix contains the information used during the cross case analysis of the interview data. The information displayed in the cells of the matrix aim to explain the idea behind the corresponding code in the transcribed interview. Next to each entry is the corresponding code number, which is used as reference during the discussion of the interviews in Chapter 4.

Criteria	Company 1		Company 2		Company 3	
<b>Methodology</b>	Choose an existing application from the platform and modify it to work in the current system	30	There is not always time to follow a formal methodology "It takes forever"	18	Big companies are able to benefit from formal methodologies. It seems that formal methodologies are aimed more a bigger companies	27
	A methodology gives the development process structure and makes it manageable. It is important to have structure and manageability during development	32	Big companies have very strict procedures that they follow and this causes them to take a very long time to get things done	18		
			Use an in-house developed methodology	14	There is not enough time to use a formal systems development methodologies	14
	It takes a while for the developers to accept a new methodology	55	Uses a very informal methodology	14	When using a methodology, the development is always successful	14
	Uses PMBOK management methodology	97	Methodologies must be used to ensure a successful system.	20	A company should develop its own methodology depending on the company character	26 27
	The methodology should be as simple as possible. This helps to simplify the development process and decrease the development time	108	A relaxed atmosphere is conducive to productivity.	20	The system development methodology adapts to the current project	33
			Whether a methodology is used is dependant on the type and size of the project as well as the client	20 21 30	Used an in-house methodology	26
	When using a methodology, the development is always successful	87	A formal methodology should be used for big projects. These are systems where you work with money or sensitive data	16	try to use systems development methodologies as far as possible	14
			Formal systems development methodologies only work in theory, not in practice	16		
			try to use a systems development methodology as far as possible, but we're not to strict on the procedures	16		
<b>User Requirements</b>	The user specification can be used to create a flow diagram	12	The user requirements get drawn up in the specification document and apply it to our network	14	They have the user write down what he wants. That is then translated into a technical document and sent to the user for approval	13
	Some times the initial user requirements you receive from the user are incomplete. Often consisting of only one sentence	29 10	Get it approved by the users	12		
			Gathering the user requirements is a very informal process	14	The technical document is applied to our engines	17
	The user requirements get drawn up in the specification document	29			Work on the actual development starts before user requirements	18

	The specifications gathered during user requirement analysis is used to determine whether the project is successful	56			analysis have finished	
	Spending more time gathering user requirement ensures a better product	87			Getting user requirements is a difficult process	12
	Get thought patterns of users.	10				
	How does the current system work	10				
<b>Design</b>	The specification is used to determine how the new system will effect all the current systems in the department	86	Not always time for the design phase	40	The platform used to develop systems is modular and is divided into functional blocks	9
	The business rules of the company and departments are taken into account during development	67			Physical distribution of the system components should be taken into account when designing the system	11
	The user specification can be used to create a flow diagram, showing the key parts of the system and how the system interacts with other systems.	12				
	Dividing the project into small parts makes the development much more manageable	73 91				
	See if there is an existing application in place that can be reused by making slight changes to the code	30				
<b>Development</b>	System: integration needs to be taken into account when developing a system	71	The standards that are available make development a lot easier	6	Some elements of development stay the same for all projects. It's just the implementation of the different elements that differ	34
	Ideal would be if there was a group of developers that could work purely on development. Once that project is released, another set of developers will work solely on maintenance	35	Because of the speed of development you need to build the system and fix any problems afterwards	40	Code generation is always the same.	
	Control is very important	98	Strict development procedures take a long time	18	Coding is an easy part of software development	40
	Before development starts, it must be estimated what the development time will be	90	You have to develop quickly	10	When deploying a system a UAT must first be performed on the testing environment. If the system gets UAT signoff, a job order is created, and the system is deployed at a time when downtime will be acceptable (Like 12 o'clock of 2 o'clock in the morning). The system is then monitored for some time	63 65
					When a system fails, a "post mortem" must be done to determine exactly where the problem was. And it should be determined how this problem can be eliminated in the future.	67
					An important fact is that there is always a tradeoff between the amount of time spent on development and the amount of time spent afterwards on	28

					debugging.  Break the development up into groups, develop and test them separately and combine them afterwards.	24 31
<b>Modular Development</b>	Dividing the project into small parts makes the development much more manageable  Whether you use modular development depends on the size of the project	73 91  91			Break the development up into groups and test each individual part separately. Then combine the parts and do an overall testing.  The platform used to develop systems is modular and is divided into functional blocks	24 31  9
<b>Reuse</b>	See if there is an existing application in place that can be reused by making slight changes to the code	30	Try to reuse previously developed code as much as possible, This also helps with flexibility  Ready made tools are reused	26  27	You don't develop the system from scratch. You have a platform setup with ready made applications, which you just modify to suit the requirements  The company could have a library of reusable applications. Then choose the applications needed for the project and adjust the base applications to cater for each client's needs. "The platform stays the same, the interface stays the same, it is the implementation that changes."	8  9 10 17
<b>Development Platform</b>	See if there is an existing application in place that can be reused by making slight changes to the code  A company should be ready for a project. From a billing - and a systems point of view	30  5	Ready made tools are reused. These tools are developed to handle communication with the databases and SMS interfaces. When such functionality is required in a system the tools are refined and reused.	27	The company could have a library of reusable applications. Then choose the applications needed for the project and adjust the base applications to cater for each client's needs. "The platform stays the same, the interface stays the same, it is the implementation that changes."  This platform has to be very stable because you can't test and qualify your applications every time  The platform used to develop systems is modular and is divided into functional blocks  The billing application, which could form part of a system, is part of the platform. This means the billing application and interface always stays the same, and is reused in all the systems where billing is required.  You don't develop the system from scratch. You have a platform setup with ready made applications, which you just modify to suit the requirements	9 10 17  8  9  9  8

<b>Development languages</b>	VB, Delphi, .Net, HTML	51	VB, .Net C# Main language	7 8	Java, C, Delphi, XML, Pascal, XXL	48 49
			There are no differences between telecoms and other areas regarding the languages used for development	7	Pascal is mostly used to develop non core applications. User interfaces and the like. Especially graphical applications	50 51
<b>Testing</b>	It should be tested that the product can correctly communicate with the networks.	75	The product should always go through a phase of testing before it is released	12 18 30	There is a specified testing environment in which testing takes place before product release. This environment is safe controllable	25 62
	If the system is going to influence a lot of people, stress testing must be performed before rollout	76	Testing is done by maintenance after release, to ensure system functionality	28	Simulators are used for testing. This means that you are able to simulate the network on which the product will run. This has the advantage that you don't have to wait for the customer to test it for you.	32 45 55
	User acceptance testing is the last level of testing before the system is released	75	Testing is essential to ensure product quality	28		
	not only to make sure the system is working, but also for quality control	57	Tests the operational aspect of the system more than development aspects.	29		
			Test for uptime	29	Break the development up into groups and test each individual part separately. Then combine them and do an overall testing.	24 31
			Because of the speed of development you need to build the system and fix any problems afterwards	40	Closed user group testing should be done on the live system before the product is made available to all.	64
					Most of the testing takes place at the billing interface	24
<b>Support and Maintenance</b>	Developers on the system have to do their own maintenance	82	Support and maintenance guys also help with testing	28	The telecom industry is very maintenance intensive	5
	A big problem they have is the fact that developers are often taken off a project that is being developed to do maintenance on another project.	82	Testing is done by maintenance after release, to ensure system functionality	28	A support an maintenance contract is sold with the project	66
	Maintenance is done to keep the system up and running and thus keep the client happy	93	24 hour support for all their systems.	28	They have a central control from which they can monitor the systems	66
	Critical errors are solved immediately by the developers. Less important errors are addressed during maintenance days, which occur every Friday	79	Don't follow strict procedures during maintenance	30	Tools like site management and version controlling makes Maintenance easier	57
<b>Document Use</b>	The user requirements always gets documented	9	Specification Document		Less documentation requirements allows for faster development times and a more adaptable system development	30
	Dataflow diagram	12			If a protocol is developed for your in-house systems, it is always documented	35
	Specification Document				Specification Document	
<b>Dataflow diagram</b>	The user specification can be used to create a flow diagram, showing the key parts of the system and how the system interacts with other systems.	12				

<b>Specification Document</b>	This document is created following user requirements analysis and then sent back to the user for approval	10	This document is created following user requirements analysis and then sent back to the user for approval	12 14	This document is created following user requirements analysis and then sent back to the user for approval	13 17
	A specification document not only shows what the user wants, but might also show why the user wants it.	29	The specification document takes into account how the internal network of the business works and how the system will fit into that network.	14	The specification is used to setup the UAT criteria after the project is finished	20
	The big networks usually write very good specifications	31			One document is the protocol specification which describes the way the system should interact with other systems.	36 38 39
	Elements in the specification includes business rules, data flows and user requirements	68				
	The specification document forms the basis from which the system is developed.	69			The specification document takes into account how the internal network of the business works and how the system will fit into that network.	17
	The user specification can be used to create a flow diagram	12				
	Serves as a means of communicating this information throughout the business	66				
	If the user specification is insufficient, it is converted into a functional specification The functional specification goes into detail about what the system is supposed to do, as well as its relationship to other systems This is not always done	70 71 89				
<b>Tools</b>	The tools that are used depends on the project	28	The systems and tools used are standard for non telecoms and telecom systems alike	7	Tools like site management and version controlling makes Maintenance easier	57
	If the project is large enough, project management tools are used	28	99% Microsoft products, some Linux servers for firewalls and mail servers	8	Most tools used are developed in house, or is open source	52
			Software products and tools used are basically the same, no matter what industry you are in	6	Simulators as a testing tool saves a lot of development time	32 45
			Ready made interface tools are reused in the development process.	27	Commercial tools are very rarely used and is always non core	53 54
			Tools are developed in-house	27	Tools like RRD tool and MRT cheap are used for maintenance and statistic	52
<b>Adaptability &amp; Flexibility</b>	A change in standards, or a required compliances introduced by the networks or ICASA has the ability to influence all the systems and business rules in a company. Steps have to be taken that the company can be ready for these changes. The company its self has to be flexible enough to be able to adapt to these constraints	23	Try to keep the system as flexible as possible to communicate with other systems  Try to reuse previously developed code as much as possible, This also helps with flexibility	26 24  26	Very flexible company  The system being developed must be flexible to so that the company can fit in with standards and protocols set by other networks  Helps keep development time down  Less documentation requirements allows for faster development times and a more adaptable system development	30  30 42  30  30
	The time available for development can be used by many unforeseen circumstances. The methodology should try to	83				

	take unforeseen events into account				
<b>Development time</b>	Using a formal systems development methodology can increase development time	87	Using a formal methodology can increase development time	10 18 19	30
	Before development starts, it must be estimated what the development time will be	90	There is not always time to go through the whole process. Planning, pre-product testing	10	
	To keep customers happy, development time should be kept to a minimum	27	You have to develop quickly	10	28
	The time available for development can be used by many unforeseen circumstances. The methodology should try to take unforeseen events into account	83	Big companies have very strict procedures that they follow and this causes them to take a very long time to get things done. Sometimes taking a month to do something that could have taken one or two days	18 19	
	The telecommunication industry demands short development times	14 18 25	A relaxed atmosphere is conducive to productivity and decrease development time.	20	46
	One of the reasons for this is because telecommunications is a fast growing industry				18 19
	Technology changes and the systems need to be developed quickly	96			
	The methodology should be as simple as possible. This helps to simplify the development process and decrease the development time	108			
	Development speed is critical. The reason for short development times is often the networks tendency to keep the project a secret until the last moment	7			15 16
<b>Communication with other systems</b>	We are constantly linked with the big networks. This means that your system has to be synchronized between 2 or more systems.	13	Keep your system open and adaptable to be able to fit into other networks	24	30 42
	Whether the system correctly communicates with other networks should always be tested before release	75	You are dependant on other networks	22 24	42
	All the systems in a telecommunications company are integrated and influence one another	2 23			39
	A change in one system causes changes in all the systems				
	The amount of systems being developed is growing. And so to the amount of interaction between them	2			
	A change in standards, or a required compliances introduced	23			

	by the networks or ICASA has the ability to influence all the systems and business rules in a company. Steps have to be taken that the company can be ready for these changes. The company its self has to be flexible enough to be able to adapt to these constraints					
	System integration needs to be taken into account when developing a system	71				
	The systems and the way they interact are very complicated	110				
<b>Industry</b>	It is a very challenging industry.	65	One of the most difficult industries to work in	31	The systems must constantly be maintained	5
	In this industry, security is very important	53	The South African telecommunications industry is very immature when compared to other countries	3	Very big. With different facets. Customer call care centers SMS, MMS, USSD. Voice, Data	68
	The industry is influenced a lot by governing bodies like ICASA	22	IT is a cut-throat market in which every thing has to happen very quickly	10	Critical systems that influence many people	5
			Telecoms is a good industry to be in because it will always be there and it is constantly growing	35	Big barrier of entry	69 70
			There are no strict, international standards concerning protocols that are to be used. The protocols that are used are either in-house, or propriety software	23		
			The world can not function without telecommunications	31		
<b>Industry Growth</b>	The phone has evolved from voice, to data to video	17 96	Telecoms is a good industry to be in because it will always be there and it is constantly growing	31	New services and technology is constantly being developed. Where once it was just that data had to be moved, it now has to be trans coded to be viewable on different phones	71
	Has to constantly stay in control of new technology, standards and laws	24				
	The systems are becoming more and more	2				
	Technology changes every 48 hours	109				
<b>Challenges</b>	A big problem they have is the fact that developers are often taken off a project that is being developed to do maintenance on another project	82	One sometimes have to interact with systems from other companies that are less adaptable than yours There is not much one can do about this	24	One sometimes have to interact with systems from other companies that are less adaptable than yours There is not much one can do about this	42 43 44
	It is a big challenge to make sure all the backup procedures are in place	49	Any telecommunication software development company in South Africa is dependant on the dominant networks.	22 24	A possible solution is to apply commercial reasons for them to speed up communication.	47
	The industry is influenced a lot by governing bodies like ICASA and the law	22 24	And their service is not always 100%. This means you have to wait a long time for them to respond to requests		It is a challenge and a necessity to manage customer expectations	41
					Getting user requirements is a difficult process	12
	Telecommunications is a very challenging industry	65	Telecommunications is a very challenging industry (Cut throat)	31 10	It is not uncommon for telecommunication projects to	15 16

	<p>Calculating risk Quality control Procurement and even Legal aspects are all challenges that has to be faced in the telecoms industry</p> <p>Developing a system from incomplete user requirements is a real challenge</p> <p>The telecommunication industry is growing rapidly, and the systems are getting more and more</p> <p>A problem in one system effects all the other systems</p> <p>You should always be up to date with the new technology</p> <p>Developing a new system is not just about software. Hardware setup is also a challenging issue</p> <p>The systems and the way they interact is very complicated</p>	<p>29 65</p> <p>29</p> <p>2</p> <p>2</p> <p>15 24</p> <p>33 61</p> <p>110</p>	<p>You have to adapt to the way the main networks do things</p> <p>There are no strict, international standards concerning protocols that are to be used. The protocols that are used are either in-house, or propriety software</p>	<p>4 24</p> <p>23</p>	<p>have development times of few weeks Telecommunication customers allow for a very short development window.</p>	
<b>Success factors</b>	<p>When using a methodology, the development is always successful</p> <p>Cost of service to customer must be kept as low as possible 100% availability</p> <p>There are different criteria for determining whether a project is a success: - What is the return on investment on the product? - Did you deliver a quality product? - Are the users satisfied with the product? - Does the product deliver on the initial user requirements?</p> <p>Use to user requirements to determine whether the project is a success</p>	<p>87</p> <p>38</p> <p>56 58 59 61</p> <p>56</p>	<p>When using a methodology, the development is always successful</p> <p>The success of the project does not rely wholly on the product developed. The marketing of the project also plays an important role.</p> <p>You know there isn't much that when you follow traditional steps and proper project management</p>	<p>13 20 25</p> <p>25</p> <p>25</p>	<p>When using a methodology, the development is always successful</p>	14
<b>Backup Systems</b>	<p>Before deploying a system to live, one must make sure the original system has been properly backed up. And can easily be recovered. The backup system should be clear of any security threats, like viruses</p> <p>It is important to have a rollback plan in case of system failure</p>	<p>42 43 46 49</p> <p>50</p>				



<b>Security</b>	Security is very important	5	Some mention of security. That	15	The machine that the application runs on has to be secure, but the software itself is not susceptible to security threats	61
	Especially in the telecommunications industry, access control is very important	53	IF it is necessary, they would look at security issues			
	Failing to implement proper security could cost the company a lot of money	54	Firewalls on Linux servers	8		
	The importance of security is not limited to mobile telecommunication systems. Security is an important factor in all IT developments	41				
	Security is a very big issue in bigger projects with big clients.	41				
	Elements like firewalls and viruses should always be addressed.	41				
	Security is also an issue with your backup strategy. Any backups made, should be clear of security threats and viruses.	42				
<b>Standards</b>	Sometimes standards change, and you have to change with it	23	There are a lot of standards	1		
			The available standards make software development easier	6		
			Development standards are the same in telecoms as in any other IT industry	7		
			There are no strict, international standards concerning protocols that are to be used. The protocols that are used are either in-house, or propriety software	23		
<b>Architecture</b>					A single system could be a combination of applications. Like network software (routing) and management software (billing)	2
					Different software components that are situated in different positions in the network	2 3
					A system consists of thousands of small applications all communicating with each other	11
					These applications can be distributed across several physical machines	11
					Data comes directly from subscribers through the network and gets sent back to the users. Data gets changed within the system.	37 59
					Sometimes data comes from two or more sources.	

<b>Costs</b>	When a project fails the company gets billed by the network. It also causes unhappy clients which could lead to further costs	27 54 93				
<b>Feasibility</b>	A Feasibility study is done for large projects.	61				
<b>Other</b>	For a big project, a feasibility study is done before hand	61				
	When a system fails, a "post mortem" must be done to determine exactly where the problem was. And it should be determined how this problem can be eliminated in the future. Doing a post mortem check is very important.	62 63 64				