THE USE OF REFERENCE PROCESS MODELS TO CAPTURE OPEN SOURCE MIGRATION ACTIVITIES

by

ONKGOPOTSE MOLEFE

submitted in fulfilment of the requirements

for the degree of

MTECH

in the subject
INFORMATION TECHNOLOGY

at the

UNIVERSITY OF SOUTH AFRICA

SUPERVISOR: Prof. ALTA VAN DER MERWE
JOINT SUPERVISOR: Prof. AURONA GERBER

NOVEMBER 2009

Contents

List of Tables

List of Figures

Acknowledgements

I would like to thank the following people. Without your help, guidance, support and motivation I could not have made it.

- Prof. Alta van der Merwe and Prof. Aurona Gerber for believing in me through their support and excellent supervision during this study.
- My manager Hennie Bezuidenhout and mentor Thomas Fogwill for the opportunity that they gave me to study and the CSIR Vula team (Randolph Verheij, Joel Ramatlhape, Eric Basson, Laurens Cloete, Lee Manickum, Mixo Shiburi and Neeshal Munga) for providing me with all the necessary documents that I needed as data.
- My parents for showing a life long interest in my studies.
- Mrs Kagiso Chikane, Meraka centre manager for her encouragement.
- Language editor Stephni Fogwill for her good English skills.
- And last, but not least, to God Almighty for all the strength and blessings.

Abstract

South Africa has shown an increased interest and awareness of Open Source Software (OSS) in the past decade. One of the reasons for this was the support from the Shuttleworth Foundation for Open Source initiatives. Migrating to OSS is a difficult and time consuming activity that should not be underestimated by the migration team. Careful planning and roll-out procedures should be in place before one commence on this journey.

Process reference models are used in different fields to capture the generic process flow of activities. For the OSS domain, no process reference models could be found for migration purposes. Therefore, this study has as aim the suggestion of an initial set of process reference models for an organisational OSS migration. These process reference models were identified by capturing the process models for a case study that entailed the migration of the CSIR software systems and desktops from proprietary to OSS. From this set of process models, the migration processes were identified and refined to a set of suggested process reference models for organisational OSS migrations. This set of process reference models are useful to determine the processes necessary for organisations considering migrating to OSS.

The study is divided into four research questions, where the first focusses on use and value of process reference models and the second on what is already known about OSS migration processes. The third deals with key processes within an organisational open source migration (OOSM) and the last with process reference models for an OOSM.

For the first research question, the use and value of process reference models and the usefulness of utilising process reference models is discussed as well as using process models as a modelling tool to identify and capture processes. For the second research question, a summary is provided of what we know about OSS migration processes and a description about what the researcher and others have learnt about OSS, OSS migrations, process reference models, the process and its structure. For the third research question, the key processes within an OOSM is discussed as well as all the processes that took place during the OSS migration project from basic administrative processes to complex processes, from the beginning of the project until its completion. Lastly, for the fourth research question, process reference models that are essential for an OOSM and possible generic migration process models bound to reoccur are identified by the researcher and validated using a focus group discussion.

Keywords: process reference model, open source software, migration, process models, process, case study, generic

ABBREVIATIONS

ARIS Architecture of Integrated Information Systems

B2B Business-to-Business

BPEL Business Process Executing Language

BPMN Business Process Modelling Notation

CSIR Council for Scientific and Industrial Research

CPSI Centre for Public Service Innovation

DOC Department of Communications

DST Department of Science and Technology

ER Entity Relationship

GNU General Public License

HTML Hypertext Markup Language

HTTP Hypertext Transfer Protocol

IDEF0 Integration DEfinition for Function modelling

IS Information System

IT Information Technology

J2EE Java2 Enterprise Edition

LOVEM-E Enhanced Line of Visibility Enterprise Modelling

NETC Northwest Educational Technology Consortium

ODF OpenDocument Format

OSS Open Source Software.
Other alternative terms for OSS include Open source software and free software (OSS/FS), free/libre and open source software (FLOSS), and free and open source software (FOSS) (Ahmed, 2005). For the purpose of this dissertation, the abbreviations open source software (OSS) and open source (OS) are used instead of the other abbreviations OSS/FS, FLOSS and FOSS

OOSM Organisational Open Source Migration

OSI Open Source Initiative

PC Personal Computer (or Desktop)

PRM (or RPM) Process Reference Model (or Reference Process Model)

SITA State Information Technology Agency

SMMEs Small, Medium and Micro-enterprises

SQL Structured query language

TCO Total cost of ownership

TCP/IP Transmission Control Protocol/Internet Protocol

TSF The Shuttleworth Foundation

UML Unified Modelling Language

WRC Water Research Council

Chapter 1

INTRODUCTION

This study suggests a set of process reference models for an organisational Open Source Software (OSS) migration that can assist with the planning and execution of future OSS migration projects.

OSS is software that is freely available at no cost for use and that can be modified to meet a user's computing needs (go opensource.org, 2005). This type of software emerged worldwide with cost savings and security advantages as some of the driving factors (Ahmed, 2005; oss.gov.za, 2006). The rise of OSS brought a tremendous change in the Information Technology sector all over the globe. Numerous government departments, the private (business) sector and educational and non-governmental organisations (Horstmann, 2005) have migrated from proprietary software and operating systems [1] on desktop computers to OSS. The problem with OSS is that organisations want to adopt it, but struggle to plan and implement OSS migration projects. Many organisations must design their own migration processes from scratch when planning such a project. Process reference models can be a solution to this problem. Process reference models make it easier for people involved in OSS migration projects to plan and execute projects. According to Rosa et al. (2005), process reference models are used to capture the common activities, roles and resources of any process in a certain environment and thereafter adapt them in another.

Anyone that wants to use process reference models for an OSS migration project, needs to identify the key generic process models that are repeatable in each migration process. The goal is to specifically look at a case study where an OSS migration project is conducted, then identify the key processes that will result in a set of process reference models for an OSS migration project. The people for whom this set of process reference models can be useful to are those involved in the OSS migration project itself such as project managers and their teams.

The rest of this chapter is organised as follows:

In this first chapter a background of the study is provided in section 1.1, followed by the problem statement and purpose of the study in section 1.2. The

[1] The term operating system refers to the operating environment for the desktop

research questions explored in this study are defined in section 1.3. An overview of the research methods used is given in section 1.4, followed by the scope and limitations of the study discussed in section 1.5 and definitions of terms is provided in section 1.6. The chapter concludes with the outline of the dissertation in section 1.7.

1.1 BACKGROUND

South Africa has shown an increased interest and awareness in OSS, because of financial reasons and due to the great work of The Shuttleworth Foundation (Belle et al., 2003) founded by Mark Shuttleworth. The Shuttleworth Foundation continues to promote the adoption and use of OSS in government departments, the private sector, and educational and non-governmental organisations worldwide (ubuntu.com, 2004). South Africa is not alone in recognising the socioeconomic benefits and technological advancement OSS brings; other nations supporting OSS include China, Chile, Spain, Thailand, India, Brazil, France, the United Kingdom and large portions of Germany (Martinez, 2005; van Reijswoud and Topi, 2003). All these countries have been more and more active in proposing OSS-based solutions because they believe it is a way for them to become competitive in the worldwide marketplace, it will reduce the cost of IT purchases, it will inform society about the uses of OSS and it will encourage the growth of OSS industries (RedHat_Inc., 2007; van Reijswoud and Topi, 2003).

As a way to show their support for OSS, South African delegates gathered for the Go Open Source Conference held 22-23 August 2005 in Johannesburg to declare, document and sign a national Open Source policy and strategy for the implementation of Free and Open Source Software in government (Mokhema, 2005). This document was then presented to former President Thabo Mbeki and members of the Cabinet after the conference. On the 22nd of February 2007 the national open source policy and strategy was ultimately approved by the Cabinet (Barnard, 2007; go opensource.org, 2005; Mokhema, 2005).

The adoption of the national open source policy and strategy paved way for projects such as Vula[2] led by Meraka Open Source Centre at the CSIR. With this project, the CSIR prioritised OSS as a key programme with the aim to lead by example and inspire other governmental organisations, private and educational sectors to also become involved in OSS migration projects (VulaWiki, 2006a) and adopt the use of OSS applications. Other leading examples of government organisations that have accepted the adoption of OSS in South Africa and are using OSS-based e-government applications, include the State Information Technology Agency (SITA), the Department of Science and Technology (DST), the Department of Communications (DOC), the Water Research Council (WRC) and the Centre for Public Service Innovation (CPSI). SITA is the main driving force in promoting the adoption of OSS nationally, ensuring that emerging OSS projects become a success in South Africa while formulating its own migration plan (dst.gov.za, 2006; oss.gov.za, 2006; Sharma, 2003; sita.co.za, 2006).

[2]The brand name of the CSIR's migration project to OSS. Vula is a verb from the Nguni language, meaning "to open" (VulaWiki, 2006a).

Migrating to a new technology can be very challenging and OSS migrations are not in any way different (Astor and Rosenberg, 2005). OSS migrations require a good knowledge of the OSS environment and it can take a long time, sometimes even years, to complete the planning and implementation of the strategy (Astor and Rosenberg, 2005). The inspiration behind the CSIR's migration to OSS was motivated by the following reasons (Dudley et al., 2006):

1. Lower or free license costs
2. Easy access to source code
3. Total cost of ownership[3]
4. Security
5. Reliability
6. Stability.

Driven by these advantages the CSIR migration to OSS aims to present an opportunity where everyone in the organisation as well as external organisations can learn from this experience (Fogwill et al., 2007). Numerous technologically minded individuals, from beginners to experts, exist within the CSIR (such as software developers, computer programmers or even users) and they may also benefit from this migration. Some employees can learn from developing, customising and testing the CSIR's Vula desktop, while users can adapt to using the OSS desktop environment instead of the proprietary desktop (Fogwill et al., 2008).

Although recent studies have shown that OSS migrations are on the rise in South Africa, there is a lack of published documentation with regards to process reference models showing how to go about migrating from a proprietary software platform to an OSS environment (Belle et al., 2003; Pawlak et al., 2004). The reason for this is that previously these types of projects were not as widely conducted and supported as they are now. This lack of documentation was one of the inspirations for this research, enabling the suggestion of process reference models for the OSS migration project done by the CSIR, focusing on the planning and implementation of moving desktops to an OSS environment. Process reference models outlining generic organisational OSS migration processes (Pawlak et al., 2004), will hopefully assist other organisations and companies migrating to OSS in future to do so successfully. The suggested process reference models intends to make life easier for OSS migration project managers and their teams, to plan and execute these types of projects better in future (Ahmed, 2005).

The term 'process reference models' is used in this study to refer to a generic process model structure (Rosa et al., 2005). One of the major advantages that process reference models provide is "Design by Reuse", which promotes replication of existing processes that enables companies to practise their business

[3]Refers to the total cost of money, time and resources associated with using and owning the software (Kok, 2005).

functions well without having to design any of the available processes from scratch (van der Aalst et al., 2003). In this research study, process models were captured from the CSIR's migration project, followed by the suggestion of process reference models extracted from the process models to assess the reusability of OSS migration processes. Process reference models were envisioned to save time and money for large, small and medium sized organisations considering the adoption of OSS once adapted and reutilised. However, an organisation must have the resources in place in order to conduct this type of a migration. Theoretically, it is widely accepted that the application of process reference models must go hand in hand with the use of a process notation, to allow users to understand the modelling configuration and master the notation used to capture the processes (Rosa et al., 2005). The process notation which is used in this study for modelling the processes of the CSIR's OSS migration is IDEF0 (KBSI, 2006). The IDEF0 notation is a standard component widely used for the creation of process models (or process mapping). It is one of the serial set of standard reference methods for process modelling that the IDEF family of methods comprises of, created by Wisnosky and Shunk (Sun, 2006).

Apart from the fact that there is no documentation about OSS migration process reference models, the majority of interesting published information that is available about migrating to OSS is about which mistakes to avoid (Astor and Rosenberg, 2005), and some of the typical obstacles that needs to be overcome (Belle et al., 2003). Both of these articles provide useful information that can be used by organisations planning to conduct an OSS migration.

1.2 PROBLEM STATEMENT AND PURPOSE OF THIS STUDY

The problem investigated by this research is the lack of documentation of process reference models of an organisational OSS migration. There is no knowledge repository or database outlining generic migration process models relevant to OSS that an individual or organisation can follow when migrating from proprietary software to OSS. Pawlak and co-authors said that "some effort has been put for integrating them in a common tool or model, however there is no solution nor model providing a global view of the F/OSS environment" (Pawlak et al., 2004).

This study will contribute by suggesting a set of process reference models suitable for an OSS migration project. In this instance the capturing of all processes was performed during the CSIR migration to OSS via the process modelling notation IDEF0. The aim is to ensure that the resulting process reference models are unique, to inspire and serve as a guide to other organisations planning to migrate from a proprietary environment to an OSS environment. The process models must be generic so that when the second or third person replicates the same migration project he/she will get the same results. The process reference models must guarantee the reduction of the risks associated with OSS migrations projects and stop similar problems from re-occurring in other organisations during the process of migration.

1.3 RESEARCH QUESTIONS

The thesis statement is "There exists a set of process reference models to capture organisational OSS migration activities".

The following research questions (RQs) supports the argument of the thesis statement.

1.3 (a) Sub-questions

- RQ1: What is the use and value of process reference models?
- RQ2: What do we know about open source migration processes?
- RQ3: What are the key processes within organisational open source migration?
- RQ4: What are the process reference models that are essential for an organisational OSS migration?

1.3 (b) Research Objectives

The following research objectives act as a guide or response to the above mentioned research sub-questions (RQs):

- The purpose of RQ1 is to provide an understanding of what a process reference model structure is and the importance of using the process model as a modelling tool (or data gathering tool).
- RQ2 intends to investigate the process of OSS migration in detail, i.e. the steps that an organisation can perform when migrating users from one technological environment to another.
- RQ3 seeks to identify the main processes that takes place during an OSS migration and will require more attention from the beginning of the migration project until the completion of it.
- The aim of RQ4 is to reveal the possible generic migration processes that are bound to reoccur when it comes to OSS migration projects of other organisations.

1.4 RESEARCH DESIGN

This research is a qualitative study. Myers (2004) describes qualitative research as the study that "involves the collection, study, interpretation and drawing of conclusions from qualitative data". By qualitative data he meant interviews, documents, data obtained through observations, field notes and other (typically) non-numerical data. The qualitative research approach was chosen as

appropriate for this dissertation to enable the researcher to make sense of the OSS migration situation and gain a much richer understanding of the processes involved when migrating to an OSS environment, via the analysis of interviews and documents.

1.4.1 Research approach

The case study approach was suitable for data collection for this study and the environment used was the CSIR.

1.4.2 Data collection and analysis

The data gathering method most used for this research was interviewing. A focus group discussion and interviews were conducted with Vula project team members and users who had migrated to the CSIR's OSS desktop software (Vula desktop). Other methods included participant observation and non-participant observation and review of the existing literature, migration guides and publications (or previous researches) that featured organisational OSS migrations. Also, the systematic research approach by van der Merwe and Kotzé (2008) and IDEF0 notation were used for the design of process models for the OSS migration project.

1.4.3 Contribution and validation

This study makes at least two contributions. Firstly, it identifies and captures OSS migration process models. Secondly it provides a suggested set of process reference models extracted from those captured OSS migration process models. This set of process reference models is a generic process model structure which hopefully can assist project managers and teams involved in OSS migration projects to plan and execute this type of project. The process that was used to identify generic process models was a focus group discussion.

A focus group discussion was conducted with several Vula project experts (or representatives) to assess the validity of captured OSS migration process models. The Vula project experts were brought together to provide information and opinions on identifying OSS migration process reference models. The project experts were chosen based on their ability to provide specialised knowledge or insight (University_of_Texas, 2007). The validation was done to ensure that the process reference models were a representation of what happened during the CSIR's OSS migration project. However, to prove their validity these process reference models have to be tested against other case studies of the same nature in future.

1.5 SCOPE AND LIMITATIONS

1.5.1 Scope of the study

This study is intended for organisations and companies migrating to OSS only. Due to the lack of documented process reference models for OSS migration projects (Pawlak et al., 2004), this study only captured and investigated generic

process models for an OSS migration project. These generic process models were identified during the CSIR's OSS migration, resulting in a library of process reference models that can be reused by other organisations and companies embarking on the same type of project, without having to develop their own processes from scratch.

The CSIR's migration to OSS was chosen as the appropriate case study for this study, as it is one of the largest organisations to adopt OSS. Because the researcher is part of the organisation it made it easier to get access to the resources needed for the completion of this study. Suitable participants and project Vula team members could be interviewed within the organisation regarding their migration experiences.

1.5.2 Limitations of the study

This study is limited to modelling generic process models (or process reference models) of the CSIR OSS migration only and does not propose a model nor introduce the development of a new model for OSS migrations (Belle et al., 2003).

Before the process reference models can be adapted by organisations and companies migrating to OSS, they have to be tested for reusability against a similar kind of an environment at about 4 to 5 companies. However, this is outside the scope of the study and it is something that can be considered for future research; given sufficient time and resources.

1.6 DEFINITIONS OF TERMS

The most important definitions covered in this dissertation are explained:

Activity is an action intended to achieve a result (JISC-InfoNet, 2007).

Generic process model is a sub-set of the original set of process models reflecting the generic and repeatable processes for a specific domain (Browne and O'Sullivan, 1995; van der Merwe, 2005).

Model can be anything that can be used to bring clarity, consistency and technology independence to any process (Kulandaisamy, 2004).

Open Source Software or Open Source (OSS) refers to software applications that are freely available for use and can be modified or improved freely to meet users' computing needs. The software is available at no cost (go opensource.org, 2005).

OSS community a group of people consisting of developers, users and IT vendors that comes together in a central place to participate in developing and improving OSS (Munro et al., 2003).

Process Reference Models (or Reference Process Models) refers to a library of individualised process models showing the graphical flow of processes and subprocesses (activities and sub-activities) as well as their relationship within a certain environment that are documented, managed and can be reused by different organisations and companies to perform their business functions without having to design their own from scratch (Rosa et al., 2005). The term process reference model is also used in this study to refer to a generic process model structure.

Process Modelling is the procedure of developing the process model using a defined standard notation (van der Merwe, 2005).

Process Model is a tool used to show how processes (or activities) flow together in a certain business environment (van der Merwe, 2005). One can say it shows how a process is working somewhere else or one can say it is a diagrammatic tool used for representation of what is happening with a set of processes.

Process is a structured, measured set of activities designed to produce a specified output for a particular customer or market (Davenport and Short, 1990) cited by (Malhotra, 2006).

Proprietary software is a software program that is not freely available for use and cannot be adapted like OSS. It comes at a cost whereby license fees are paid to those that develop and controls the software (UNESCO, 2007).

Total cost of ownership refers to the total cost of money, time and resources associated with using and owning the software (Kok, 2005).

Vula desktop is the CSIR's version of the Linux system, modified to suit users' computing needs within the CSIR and outside the organisation (VulaWiki, 2006b).

1.7 OUTLINE OF CHAPTERS

Chapter 1, the introduction, provides a background to the study and gives an overview of the research problem, research objectives, research questions, research methodology, as well as its scope and limitations. Chapter 2 addresses the theory related to the study, that is a historical (or theoretical) overview of the OSS environment and process reference models. Chapter 3 describes the research design as well as data gathering tools and techniques used during the study to collect OSS migration process models and to suggest process reference models for an OSS migration project, including the context of a case study environment. Chapter 4 analyses the detailed factors which have led to the adoption of the OSS migration project at the CSIR and the whole process of the migration. Chapter 5 discusses the contribution made by the study, as well as the summary of generic process models. Lastly, Chapter 6 concludes with recommendations for future research identified during the study. Figure 1.1

demonstrates the Dissertation Map, graphically depicting the above specified chapters of the dissertation.

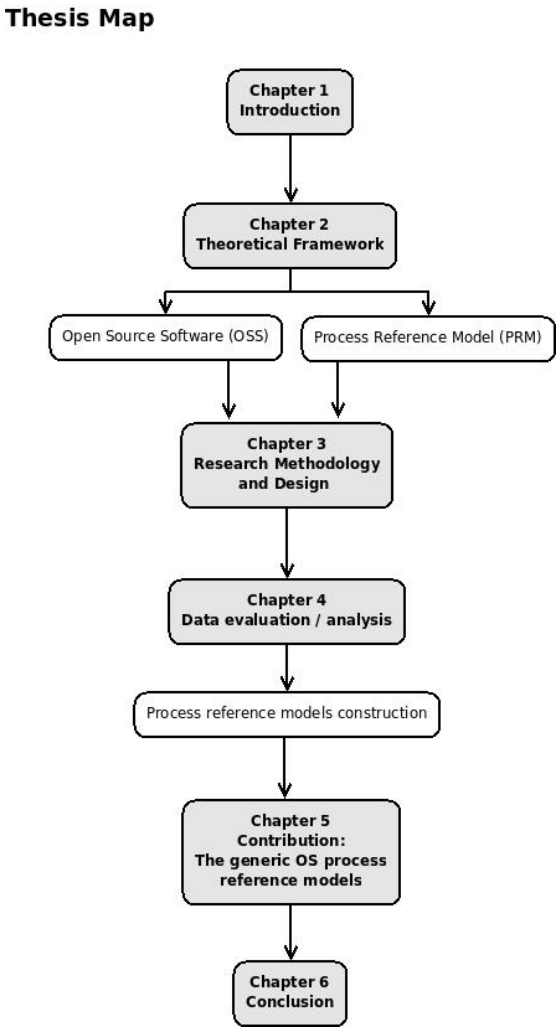

Figure 1.1: Dissertation Map

Chapter 2

THEORETICAL FRAMEWORK

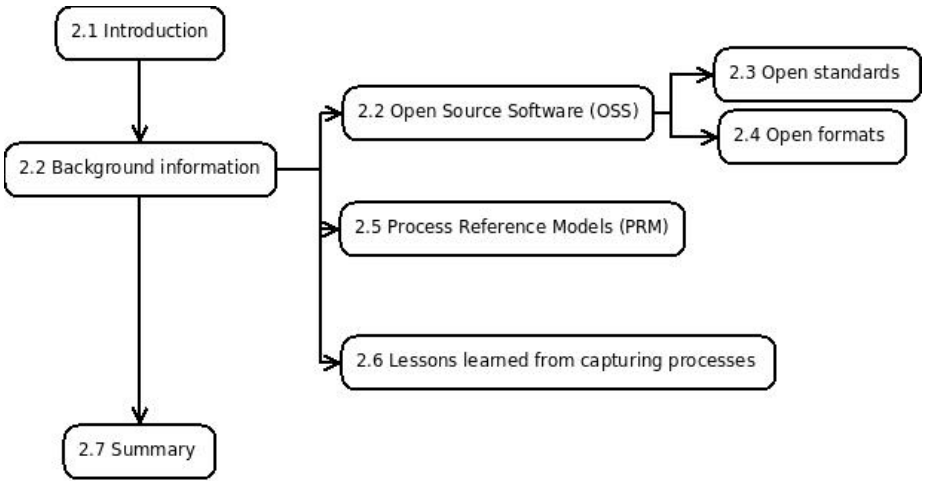

Figure 2.1: Chapter 2 Outline

2.1 INTRODUCTION

This chapter examines the theory or literature related to the study, as a response to the first and second research questions (RQ1 and RQ2 in Chapter 1, see Section 1.3) and has been organised into the following sections. Section 2.2 discusses the literature on OSS and the part it continues to play in South Africa is covered in Section 2.2.2. Section 2.2.3 provides a brief summary of the CSIR; briefly about its history as one of the leading research and scientific organisations in South Africa that develops highly recognised software. Section 2.2.5 discusses the importance of documenting activities for projects or businesses, with the focus on process reference models' structure. An overview of open standards is discussed in Section 2.3 and open formats in Section 2.4, including their relation to OSS. An overview of process reference models structure is also provided in Section 2.5. Section 2.6 addresses what has already been written about OSS (or what is already known about OSS), that means the literature on OSS and its usefulness. The chapter is outlined in the Figure 2.1 above.

2.2 BACKGROUND INFORMATION: OSS

OSS is produced by a self-organised community that engages online and discusses different ideas about how open source software can be further improved upon to meet users' needs (Foster, 2008; go opensource.org, 2005). As pointed out in the previous chapter, it is software that comes at little or no financial cost and has been licensed in a manner that allows users to study, edit, improve and redistribute it without having to pay any royalties to those that developed the software in the first place (Ahmed, 2005). This suggests that with OSS no licence fee is required for using the software, the software has been publicly published to be used and shared among users as many times they want. OSS has become very popular worldwide and it is currently used and adopted in most private companies, governmental organisations and academic institutions for variety of reasons. These are the reasons: (1) it is free to use, copy and share; (2) the OS operating systems are more reliable than other operating systems; (3) the OS operating systems are more secure than other operating systems; (4) it can be customised to suit specific business functions or users' needs; and (5) it even works on old personal computers[1] (Dudley et al., 2006).

The most popular OS operating system used is Linux, which was developed by Linus Torvalds in 1991 who at the time was a student at Helsinki University (Peeling and Satchell, 2001). The introduction of Linux brought competition to Microsoft Windows and is regarded as the replacement to Microsoft Windows and other commercially produced operating systems (van Reijswoud and Topi, 2003). Microsoft Windows is referred to as proprietary software where one has to pay for license fees for using the software (Ahmed, 2005; Dudley et al., 2006). Linux was produced as OSS that is freely distributed to compete with proprietary software that is offered at monetary value. Like Microsoft Windows, Linux is used and adapted by several companies and organisations for desktops and server environments (Peeling and Satchell, 2001). Figure 2.2 depicts different

[1]Small desktop computer designed to be used by one person at a time.

types of open source operating systems, while Table 2.1 shows common proprietary software programs together with their OSS alternatives (Peeling and Satchell, 2001).

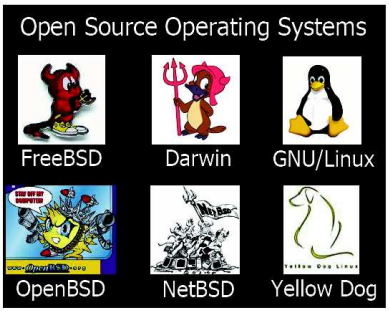

Figure 2.2: Examples of open source operating systems

Purpose	**Proprietary Software**	**Open Source Software**
Operating system	Microsoft Windows	Linux
Web browser	Internet Explorer	Mozilla Firefox
Office suite	Microsoft Office	OpenOffice.org
Image editor	Adobe Photoshop	GIMP

Table 2.1: Examples of proprietary software and their OSS alternatives

2.2.1 OSS and its components

OSS is considered somewhat equivalent to free software[2] and can be distinguished by three key components, as described by Simon (2005):

1. The first component that defines OSS is the **community**, which comprises of developers, users, and IT vendors playing a key role of coding, testing, and providing support for OSS projects (Simon, 2005). Munro et al. (2003) defined OSS community as consisting of those who participate in developing and improving OSS, while Dibben (2004) observed it as a group of people, usually like minded-people, who come together online to participate, debate and share information. This community uses a range of online tools like discussion forums, chatrooms, discussion email lists, and Information Management Systems (WebCT) to work or chat.

2. The next component defined by Simon (2005) is the **development methodology** which uses a community and peer review approach to develop soft-

[2]Free software is software that is freely available to the general public for redistribution, modification, examination or any other conceivable purpose without restrictions (Allison, 2009; fsfe.org, 2009).

ware. The code is openly published for everyone to see or review, to decide whether it is good or bad and to then offer feedback on how to improve it.

3. A **licensing approach** is the third component that provides free access to source code and conforms to one of about 60 licenses authorised by OSI[3] (Simon, 2005). All of these licenses allow users to view and modify the source code but the conditions that specify code modifications and extensions vary greatly.

OSS does not only comprise of these three components. OSS offers much more and numerous articles were published on OSS, about its advantages and disadvantages by Ahmed (2005), Dudley et al. (2006), Foster (2008), Simon (2005) and Dalziel (2003) and many more.

Cerri and Fuggetta (2006), Dalziel (2003) and Simon (2005) also identified the characteristics of OSS which include those shown in Table 2.2.

OSS characteristic	**Description**
1. Transparency	Transparency means that nothing is kept confidential, in this case OSS workings are exposed to the public and can be modified by anyone.
2. Open standards	Open standards is a standard not owned by a single company. It allows anyone to use the standards as well as contribute to its further development (Simon, 2005; Texas_Department_of_Information_Resources, 2008). It is publicly available rather than being kept confidential. Open standards promote interoperability between systems; that means sharing of information among systems. Both proprietary software and OSS can use open standards, as long as their products conform to those standards.
3. Interoperability	Interoperability defined by Dalziel (2003) and Simon (2005) as the ability of different systems to share information in such a way that both can act in an equivalent manner on the information that will result in equivalent user outcomes. It emphasises the importance of openness among systems when exchanging information with the considerations of security, privacy and business rules in place to enable this effective interaction.
4. Customisability	Customisability allows the modification and improvement of the source code in an OSS environment. This implies that the source code together with its technological specifications can be altered to meet users' needs. This characteristic of OSS allows many software applications to be customised and redistributed in their modified form.

[3]OSI is a non-profit corporation dedicated to managing and promoting open source software (opensource.org, 2009).

5. Open licensing	Open licensing in general terms means that users do not own software; rather they obtain a license to use it (Cerri and Fuggetta, 2006). The licence defines the terms and conditions for using the software. For instance, OSS is not unlicensed but rather specifically licensed to allow and encourage wider use, distribution and modification of its non-confidential source code.

Table 2.2: The characteristics of OSS

Not only Cerri and Fuggetta (2006), Dalziel (2003) and Simon (2005) reported on these characteristics of OSS. Hoe (2007), Opensourceafrica.org (2007) and GoOpenSourceTaskTeam (2003) also stipulated the technical and financial benefits brought forth by OSS, impacting positively on the economic development of nations. It is these benefits in Table 2.3 (that Hoe (2007), Opensourceafrica.org (2007) and GoOpenSourceTaskTeam (2003) observed and reported on) that persuaded organisations and companies to consider the adoption of OSS.

Direct benefits	**Description**
1. Cost savings	Due to costs of proprietary software licensing, which are often hidden, OSS adoption can assist with cost savings as it is freely available to all.
2. Application mix	When it comes to OSS, many useful OSS add-on programmes are available for free. These add-on programmes can expand or improve the operations of a small organisation. These useful add-on programmes include amongst others, project management utilities, graphic design programmes for creating stationary, product packaging and signage, and sophisticated multimedia tools.
3. Recycle old desktops	With OSS the productive life of an older desktop can be extended, because OSS comes in many varieties and some of its installations can even run on 10 year old desktops.
4. Business-to-business transactions	For those organisations needing to connect directly to the computing systems of their partners or other organisations they are collaborating with, OSS offers an open platform, compatible with all major interconnection standard protocols and middle-wares.
5. More robust and reliable software	OSS's robustness and reliability provides less vulnerability to viruses. To prevent a loss of information from occuring, OSS gives confidence that users' information is free of security threats such as viruses and software bugs, that can become costly to fix (Hoe, 2007).
6. Learn, redistribute and enhance	As specified previously OSS provides everyone with the ability to modify and customise source code freely. It promotes an environment for positive competition, self-learning, exploring and cooperation (GoOpenSourceTaskTeam, 2003; Hoe, 2007).

7. Prevent single vendor and technology lock-in	Not everyone is technologically oriented, some users depend on software vendors to provide them with products that will meet their needs, while others prefer developing and modifying applications suitable to their own needs. OSS provides this support by reducing vendor dependence and lock-in of proprietary software into less costly, securable, customisable software that can be designed to meet users needs (Hoe, 2007; Simon, 2005).
8. Promote and preserve open standards	Stated as one of the OSS characteristics, open standards is associated with various principles or rules that products and vendors need to conform to when using the software. They are published rather than kept confidential and can be accessed by anyone following the standard.
9. OSS benefits poorer society	OSS provides affordable and accessible software to run on old and new desktops (Hoe, 2007). Not everyone can afford commercial software that sometimes come with hidden costs and OSS is therefore an alternative (Martinez, 2005).

Table 2.3: Direct benefits of adopting OSS in organisations

After realising that many authors only focus on reporting about OSS's financial and technical-oriented benefits, the GoOpenSourceTaskTeam (2003), Martinez (2005) and others further identified socioeconomic (or developmental) benefits. These socioeconomic benefits are important to consider when evaluating the proper place for OSS adoption in the developing world. The suggested socioeconomic benefits that OSS provide is listed in the following summary:

1. OSS puts users' needs first.
   OSS puts users' needs first, by having in place a community that understands users' needs and knows that not all commercialised software meets every user's needs or can be afforded by everyone. By being free (or publicly available), secure and customisable the software meets users' computing needs (GoOpenSourceTaskTeam, 2003).

2. OSS promotes transparency and accountable government.
   Governments are utilising OSS not only because they will be saving costs on paying software licenses, but because they understand that OSS solutions are the best in terms of stability, scalability, flexibility and efficiency Martinez (2005). Therefore the usage of open technologies such as OSS and open standards promote a culture of openness and transparency in government as well as in society. OSS facilitates sharing of information (interoperability) among government systems, its stakeholders and the society as a whole (Fuggetta, 2004; Simon, 2005).

3. OSS supports the local IT industry and digital self-sufficiency.
   OSS is concerned with ICT spending in local companies, which is why it provides several ways of minimising the costs (see Table 2.3). The main reason why OSS is encouraged in local companies, is to educate and train company workers about OSS. This training will produce valued, employable and skilled workers with OSS knowledge, at the same time ensur-

ing that the knowledge is shared from the young to the older generation (GoOpenSourceTaskTeam, 2003).

4. OSS supports entrepreneurship and business formation.
   The shift from proprietary software to OSS "brings technology to everybody and empowers people, communities, governments and companies" (Martinez, 2005). It furthers the success of small, medium and micro-enterprises, by creating opportunities for the entrepreneurial success of SMMEs as well as jobs for those wanting to learn more about OSS, at the same time leading to economic empowerment.

5. OSS supports innovation, local solutions and learning.
   OSS encourages hands-on, self-directed, experimental learning of source code, resulting in software solutions that are particularly suited to meet users' computing needs. This an empowering way of teaching people nationwide or around the globe (Fogwill et al., 2007).

6. OSS promotes collaboration and open standards.
   OSS encourages and self-supports open standards. As previously specified in Table 2.2 open standards are a set of standards, not controlled by a single company, that any interested party can access, use and contribute to (Dalziel, 2003; Fuggetta, 2004). Examples of open standards are HTTP, HTML, TCP/IP, XML and SQL, just to name a few (Cerri and Fuggetta, 2006; Texas_Department_of_Information_Resources, 2008; West, 2004).

7. OSS reduces vendor dependence and lock-in.
   OSS aims to prevent the single vendor and technology lock-in users have with proprietary software. OSS provides this support by encouraging users to have a clear understanding of the OSS environment and utilise the benefits it brings (Krechmer, 2005; Simon, 2005).

8. OSS allows market entry for firms that cannot withstand corporate competition.
   The support for OSS by emerging companies in the ICT industry balances the culture of market competition. Many of these emerging companies choose to go the OSS route because its less costly and it is customisable and securable. This culture helps with social upliftment and economic development (GoOpenSourceTaskTeam, 2003; Martinez, 2005).

9. OSS raises the profile of South Africa in the global economy and narrows the digital divide.
   Any country besides South Africa participating in OSS raises the profile of the developing world. This particular country helps to "demonstrate its capabilities and its desirability as a progressive, technologically literate and knowledge-savvy nation" (GoOpenSourceTaskTeam, 2003).

OSS does not only consist of these positive benefits, it has been identified that there are also some disadvantages that come with this type of software such as the lack of support by vendors, difficult installation instructions, hardware compatability problems, lack of technical skills, user resistance and warranty issues

(Thomas, 2005). However, none of these disadvantages have discouraged organisations and companies within South Africa and other countries from adopting OSS (GoOpenSourceTaskTeam, 2003).

2.2.2 The impact of OSS in South Africa

Not only is OSS finding a place in South Africa, it has also gained prominence in other nations world-wide (Martinez, 2005; RedHat_Inc., 2007). With the expertise that is made available by the OSS community, all the nations that have begun to take advantage of the opportunities offered by OSS, including South Africa, are greatly benefiting from using and adopting OSS (van Reijswoud and Topi, 2003). After realising the benefits that OSS brings, South Africa took the initiative of introducing OSS migration projects within its governmental departments with the aim of cutting costs. The adoption of the National Open Source Policy and Strategy was introduced by Cabinet earlier in 2007. This decision was supported by numerous governmental organisations, including the CSIR, SITA, DST, WRC and CSPI. All these organisations embarked on the planning and implementation of using OSS on their desktops and servers[4].

According to Dudley et al. (2006), using OSS within any organisation needs a thorough evaluation because experience in the IT sector has shown that everything that is free is not necessarily good. Sometimes there are hidden obstacles and costs involved when shifting from one technology to another, in this instance to OSS (Astor and Rosenberg, 2005; Dudley et al., 2006). No one can guarantee that the movement to OSS will be easy. Not everyone likes change. Therefore migrating from a proprietary software desktop to an OSS desktop (or Microsoft Windows to Linux) can potentially bring disruption to every user in your organisation (Astor and Rosenberg, 2005). Despite this, many organisations including the CSIR are still committed to making this move because they believe that like any other form of migration, OSS migration has its challenges but is still doable.

Previous web searches reported an increase in the number of government departments, private sector, academia and non-governmental organisations in South Africa that are leaving proprietary software for OSS due to high costs of license fees associated with proprietary software (Dudley et al., 2006). The CSIR migration to OSS was motivated by several factors: (i) to be one of the largest organisations to have adopted OSS in South Africa, (ii) to share the knowledge of the migration process acquired (or data collected) during the project with the public instead of keeping it confidential, (iii) to prepare other organisations to plan and execute better OSS migration projects, (iv) to remove the fear and uncertainty associated with the adoption and usage of OSS, (v) to empower users and scientists, (vi) to foster local ICT skills development and (vii) to further socioeconomic development (Mokhema, 2005). The factors above are what makes the CSIR migration to OSS unique when compared to other organisational OSS migrations. The completion of the CSIR OSS migration will provide other organisations with the knowledge repository or relevant database of migration process models. These process models will be used for the extraction of

[4]A server is a computer or a program that provides services to other programs or users, in terms of having access to files and printers as shared resources on a computer network

process reference models, which aims to assist organisations to plan and execute their OSS migration projects better in future.

2.2.3 Migration projects

A Web search for articles on the migration from proprietary software to OSS was conducted with regards to reviewing some work already done by migrated organisations. The results of that web search were several case studies found on general migration projects involving organisations that had moved from a proprietary distribution to an OSS one. The results are depicted in Table 6.1 in Appendix B.

Soon after the CSIR's migration project was launched in June 2006 after the CSIR President Dr Sibisi made the decision of migrating the majority of the organisation's desktops to OSS (Linux in this case), it turned out that the CSIR was the largest firm to have ever embarked on this kind of a project citepOSS-DeclarationConference. It was envisioned that this project will not only benefit the CSIR, but it will also act as an opportunity to educate emerging young software developers in this field. The OSS route offers many software developers the flexibility to develop and experiment with innovative new functionalities and the opportunity to collaborate with other open source communities while at the same time encouraging them to create their own open source communities in alignment with their projects (Ahmed, 2005). Another motivation was saving the increases of annual licensing fees (Fogwill *et al.*, 2007). Other reasons can be viewed in section 2.2.1. The decision made by Dr Sibisi received support from the CSIR Executive Committee and the rest of the CSIR employees and external stakeholders that CSIR collaborates with, such as SITA and DST citepOSSstandardsTalk.

According to the team that was appointed to lead the CSIR migration project 'Vula', the CSIR embarked on this project because it believed that through this project it will "continue to provide better support for the core science and technology activities of the organisation" (Fogwill et al., 2007; VulaWiki, 2006b). The aim is to provide insight to the planning and implementation that project leaders and teams can use to set up and execute OSS migration projects in future. The incorporation of these insights into other migration projects will help project teams to reduce costly mistakes and wasted time (Ahmed, 2005).

Dudley et al. (2006) described the rationale behind the decision made by several organisations for making the shift from the proprietary to an OSS environment. They outlined the plan, benefits and challenges faced by these organisations when deploying the open source distribution. They also explained that the deployment of the OSS in these organisations were aligned with cost considerations and the freedom of learning and adopting effective new technologies which were open source related.

Ranganathan *et al.* cited by Ahmed (2005) identified four main challenges that can be encountered by organisations when shifting to new distributions in desktops. They said that using a new distribution in desktops, be it an OSS or proprietary one, can bring about: (i) changes to processes; (ii) adjustment to

new systems, which will involve more human interaction (e.g. data entry), (iii) involve training staff, and (iv) shifting users mindsets towards the distribution installed in new computers. These are some of the challenges that are bound to be experienced by organisations when migrating to an OSS environment, because the migration project can be a quite challenging process (Belle et al., 2003). Not everyone within the organisation will be supportive of the migration or support the project. Therefore it has been reported by Ahmed (2005) and the Novell-Versora-Linux-Team (2005) that users can be a bit unsettled when it comes to migration. Meeting their technological demands and preparing them by making them comfortable and skilled for that specific distribution can be the most problematic aspect of migration. Ahmed (2005) went further to say that "training may help but it may not increase the user level of comfort or skills".

Fitzgerald and Kenny (2003) learned two main obstacles during the course of migrating to an OSS: (i) changing users' mind-sets when adopting the usage of OSS solutions (this is similar to what Ranganathan *et al.* mentioned above); (ii) resistance from users who feared the discomfort of becoming less skilled by moving away from a popular proprietary system. They reported all these obstacles as being related to human behaviour. This can indeed be one of the most critical challenges that could be experienced by project leaders and teams during the migration from proprietary to OSS.

Ranganathan *et al.* (2004) as cited by Ahmed (2005) continued to report on the six lessons learned about bringing transformation within an organisation. He explained that in order to allow transformation to take place within the organisation the top management, that is executives, managers or project managers, must first be the architects of change and lead by example to allow individual transformation to precede organisational transformation. Without that in place, Ranganathan *et al.* (2004) as cited by Ahmed (2005) and Novell-Versora-Linux-Team (2005) believed that the rate of change will not match the rate of acceptance of the new technology. There must be diverse change agents such as organisational leaders (or top management) placed to first sense, recognise and respond to transformation themselves. The same goes for OSS migration projects, they bring about transformation within an organisation, but such a transition must first be consumed and adapted by organisational leaders.

Migration projects can be huge and expensive (Belle et al., 2003), especially when one is not familiar with the process to make it a success. Therefore, the costs of migrating should be given as much consideration as the cost of development and maintenance, especially when it is difficult to develop new applications that will meet users' needs. Sometimes even to migrate data can be difficult (Ahmed, 2005).

Kovács and Kochis (2005) explained that the responsibility of organisational leaders such as OSS project managers is to calculate and evaluate risks associated with the migration to OSS within their organisation. They identified the three sources of risk as: security, cost and users. These three sources were considered a major risk in a proprietary environment due to various reasons. One reason is that security threats such as viruses and software bugs can become costly to fix if experienced. Another reason is that these risks may affect users'

productivity within the organisation. With OSS users are less likely to experience security threats and viruses. Lineweaver (2003) together with Kovács and Kochis (2005) as cited by Ahmed (2005) emphasised the need for adopting OSS, as a way to assist organisations with handling these risks and with finding opportunities that will allow them to save money through the usage of OSS.

In summary, Weiss (2005) and KBSt (2003) cited by Ahmed (2005) reported on the substantial key factors that leads to the success of several migration projects. KBSt (2003) as cited by Ahmed (2005) defined the success of the migration project as when the "desired aims and results for all stakeholders are achieved within the planned and agreed time and budget frames". Horstmann (2005) agrees, but explained that at other times it is possible to realise at a later stage that one has forgotten some of the important migration steps (or another order for the migration steps) that would have been necessary for successful results of the migration project. Sometimes, this might be because of lack of information or skills which should have been acquired earlier in the migration project. Hence it is important to provide suitable training for everyone within the organisation that is involved in the OSS migration project. According to Horstmann (2005) "when following the proposed migration workflow, one can be sure that every aspect of the task is being taken care of and that dependencies are observed". The above mentioned provide the motivation for this research which is to suggest process reference models for an OSS migration project. To guarantee project leaders and their teams of all the necessary steps to follow when making a shift from proprietary to OSS, a good leadership style, time and effort are required from the project leaders and the team to make critical decisions regarding the project and to implement them. Weiss (2005) reported on traditional factors that had nothing to do with OSS migrations that determines the success of open source projects. His main focus was on open source communities and not migrations. He said that what makes an open source project successful is the number of active community members in that specific open source project and the number of downloads made from that project will measure the popularity of the project on the internet.

2.2.4 Factors to consider regarding migration to OSS

Leading companies and organisations confirms the fact that OSS lowers the total cost of ownership and outperforms proprietary software Microsoft Windows (Kok, 2005). Therefore, ICT project managers and practitioners planning an OSS migration for an organisation must consider the following types of factors and questions suggested by van Reijswoud and Topi (2003) before planning for the migration can occur. These are similar types of questions that the Vula project team members asked themselves when preparing for the CSIR migration project to OSS. Those factors are:

1. Examine true benefits.
   It is highly recommended that top management and ICT technical staff take a look at the benefits that comes with the move to OSS. Certain questions must be asked when considering the move to OSS, such as the following: What are the business benefits? What are the financial returns? What are the hidden costs? One recommended practice is to first

look into the cost of proprietary licensing and analyse it over time, and possibly get the software vendor to explain it.

2. Infrastructure audit.
   In this instance the question that will arise is: What are our core business practices? It is recommended that organisations do a quick audit (or survey) of the most common tasks performed on desktops. This quick survey is sometimes done by means of a questionnaire and includes the number of desktops, software utilised by users on the desktops, network servers and peripherals (printers, scanners etc.) used in the organisation. This information will help skilled ICT technical staff to better identify OSS alternatives for the currently used proprietary software and choose open source equivalent applications fit for the scale of the migration.

3. Phased approach.
   Many companies, when migrating to OSS start with network servers before moving to desktops. It is therefore critical to first identify key workgroups (or power users) within different departments such as sales or accounting that can easily be migrated. This way the project team will be able to determine a running order for the migration of users.

4. Staff buy-in.
   Not all staff members will be equally accepting of change, some might resist. Hence the importance of openly discussing the migration to OSS and the cost savings and security advantages that will be achieved by this move. This is an effective way of exposing the potential resistance ahead of time and addressing everyone's expectations. It is ideal that the organisation appoints one or more OSS champions to hear users' grievances and at the same time effectively promoting the advantages of moving to OSS.

5. Change management.
   Change management is also in line with the open communication channel among the staff members and top management. It is possible that during the migration project downtime can be experienced (users not being able to utilise their desktops due to certain errors they will be getting on their desktops). This can gradually cause a loss of productivity for users and the organisation as a whole. To balance this, an open communication channel is required. Staff training is also essential to ensure no loss of productivity (see the point below for more on training staff).

6. Support for organisation.
   It is important to upgrade the skills of current ICT technical staff and users to keep up with the business needs of an organisation. Even though sometimes it is technically possible for relatively unskilled staff to install OSS applications on their desktops, staff training is required and should be planned and costed ahead of time. It is possible to find an OSS support

firm that can assist with the support, but bear in mind that outsourced technical support may become costly over a long period of time.

2.2.5 Capturing OSS structures

Documentation of processes are efficient for reference purposes (Barn, 2007; Childe et al., 1997). Therefore every organisation or business that has business needs, functions and goals, must capture its processes at the initial stage of its operation (Dortch, 2009). Any type of business or project which operates with nonexistent processes or undocumented common practices may eperience a loss or failure of that business or project. This was observed by Tyrrell (2000) as one of the greatest problems facing most organisations, even emerging ones, hence the importance of documenting (or capturing) processes is often emphasised.

It is necessary for businesses to document the processes which are commonly done in a repeatable manner, particularly those that are of benefit to the organisation. Documenting repeatable processes (or ways of doing things), for an open source project or any other form of business will help people because they can refer to saved records (archives) that will allow them to learn from the experiences of those who have already gone the same route.

To capture a process simply means to write down various activities that need to be produced and the order in which they have to be planned or organised, produced, tested and documented to ensure that they meet the requirements (Tyrrell, 2000). Once a process is defined and captured, there is less of a chance of repeating mistakes if the process is done the same way every time. The more a process is repeated, the bigger the chance that it will be refined constantly. Again it is easier to train people if the process is documented and consistently performed each time (Turbit, 2005). Doing things the same way each time, makes the process more efficient and easier to reuse (SmartDraw.com, 2009; Tyrrell, 2000).

The same rules apply to the documentation of the OSS migration activities. The CSIR and its Vula project team saw the importance of documentation because they realised that with the documentation of their organisational migration activities (or experiences), they can assist other organisations with the knowledge of conducting these types of projects. The research outcomes of this study are to be of help to ICT project managers and their teams when they consider the adoption of OSS. Process reference model structure will play a role in ensuring that generic process models of an OSS migration project are identified and captured (Refer to the detailed Section 2.5 about process reference models).

Still about OSS, the next section will focus on open standards because OSS and open standards are connected. More will be revealed during the discussion in the next section.

2.3 OPEN STANDARDS

Open standards are relevant to this study because like OSS they are used both in a proprietary and an OSS environment. They both have similar benefits such as transparency and interoperability and can be used by everyone (Cerri and Fuggetta, 2006; Texas_Department_of_Information_Resources, 2008).

According to Fuggetta (2004) and Perens (2007), an open standard is a standard that is not controlled by a single company. It is a standard that is publicly available and has various rights to use associated with it. He further reported that open standards can be adopted in both environments, the OSS and the proprietary software one. In practice, this means that open standards can be implemented by both commercial systems and open source systems, provided that all systems conform to the standards (Dalziel, 2003).

In the report generated by the Centre_for_Ecological_Sciences (2009) and Ghuha as cited by Shewale (2009), the terms 'open' and 'standard' have different meanings associated with their usage. The term 'open' is restricted to royalty-free technologies while the term 'standard' is restricted to technologies approved by formalised committees that are open to participation by all interested parties and operate on a consensus basis. IBM (2006) identified several examples of open standards which include file formats such as HTTP, HTML, TCP/IP, XML and SQL. These sets of file formats (or open standard examples) are available both in proprietary and open source operating systems such as Linux and Microsoft Windows (Fuggetta, 2004). A user can run them both on a Linux desktop or even on a Microsoft Windows desktop without experiencing any compatability issues. For example, the benefits of using HTTP or HTML have been demonstrated on a large scale, the same applies for other open standards (Simon, 2005).

Dalziel (2003) explained that it may be natural to think that open source would be preferable to open standards if you were forced to choose between them. However, Shewale (2009) reported that an open standard is sometimes coupled with open source with the idea that a standard is not truly open (or free) if it does not have a complete open source reference implementation available. This is because in an open source software development environment, all of the source code is freely available and if it does not correspond to open standards, it can be modified to solve these standard-compliant issues (Dalziel, 2003). He went further to say that in the case of commercial systems which support open standards, these commercial systems rarely provide access to their source code, so external developers are not able to change the software as desired. Thus the open source alternative appears to be a more secure and customisable option.

Because open standards are used to specify formats, they are sometimes referred to as open formats (Shewale, 2009). It must be encouraged that proprietary software interacts with the rest of world by using open standards and open formats (open formats are discussed in more detail in Section 3). Again, this is not the same as requiring proprietary software to be open source, it means that open standards can be utilised both in a proprietary and OSS environment, as open formats.

2.3.1 How open standards have emerged from OSS

Open standards is a popular topic of discussion among governments and other organisations (Simon, 2005). Many studies by governments and by IT analysts indicate that OSS and open standards are inherently valuable (Dalziel, 2003). For example, the CEO of the CSIR Dr Sibisi and others have presented talks on open standards at Govtech (VulaWiki, 2006b). This goes to show that South Africa is not alone in recognising the benefits that comes with OSS and open standards.

According to Krechmer (2005) and Simon (2005) many software capabilities began as OSS solutions, but with time these OSS solutions evolved into open standards which are positively affecting governments and organisations world-wide today. He suggested that to realise the advantages brought forth by OSS development, one must understand how open standards have emerged from OSS and evidence has shown that open standards have emerged due to the following three significant reasons:

1. Development of a high-quality product that became trusted by many users
2. Competition of vendors
3. Demand from users.

Simon (2005) explains that the example that supports the above is SENDMAIL which is a standard program for Internet mail transfer. He says "SENDMAIL was developed by Eric Allman in 1981 at the University of California, Berkeley during a period of proliferation of networks and e-mail protocols. SENDMAIL was the first software of its kind used to complete mail transfers, it did not reject incompatible mail from different networks, instead it modified the mail so that it could be transported to its destination" (Simon, 2005). With interest in SENDMAIL growing rapidly, in 1986 Eric Allman decided to cease his development of SENDMAIL, giving vendors and other developers an opportunity to enhance his work regarding SENDMAIL capabilities. This led to the release of two additional versions of SENDMAIL with special features.

In 1989, Eric Allman continued his development of his SENDMAIL version. The existing competition by vendors and developers enabled Allman to incorporate some excellent features from the released versions (Simon, 2005). Because many users trusted and used the capabilities of SENDMAIL, they began to demand from vendors and developers that specific features be added similar to those of Allman's software. As a result of this trust, competition, user demand and facilitation by the development technique, OSS emerged and SENDMAIL was declared the standard for mail transfer. This means the needs of customers and the abilities of developers, and not solely the output of proprietary vendors, shaped the Internet into its current success, explains Simon (2005).

Besides SENDMAIL, there are other several OSS solutions that have been as successful as SENDMAIL such as Linux, the Apache Web server, Berkeley Internet Name Domain (BIND) and Mozilla. All these OSS solutions have become open standards with the help of the superior OSS development technique and through the support of the significant reasons mentioned above by Simon (2005).

2.3.2 Open standards vs OSS

The difference between OSS and open standards is as follows:

- The term open standards means "standards that are created and maintained in an open manner, using a democratic approach, where no single individual or company controls the standard" (Bulusu, 2003). The most important qualities of open standards are that they are publicly available to all and generally no royalty fees or fees for creating an implementation conforming to the standard is payable (Bulusu, 2003; Fuggetta, 2004). With open standards, vendors are allowed to create either OSS or proprietary software, as long as it conforms to the standards (Bulusu, 2003; Dalziel, 2003).

  According to Cerri and Fuggetta (2006) open standards are released by bodies that have the power to control the definition of a standard. While Fremantle (2003) argued that in his experience, "open standards groups are not always effective at creating new stuff instead they excel at tightening up already created stuff". He went further to say that many open standards groups consist of companies who are in strong competition and that open standards creates an open environment for such competition among these companies. The aim is "to allow them to agree to conformance, open up the market and grow the potential business through standardisation of software products" (Fremantle, 2003). The web site openstandards.org was created to hold a number of web pages for groups producing open standards and also to serve as a repository for standards relevant to the open source community (open std.org, 2002).

- OSS is used to refer to "software whose source code is freely available to be shared by anyone and can be modified, subject to the terms of a licensing agreement" (Bulusu, 2003). It is developed by like-minded people who want to share the effort of developing code and share the results (Fremantle, 2003). In his experiences Fremantle (2003) reported that the result with OSS are usually much more creative and expansive than those of open standards.

2.3.3 Prominent open standards

As previously mentioned, some of the well-known open standards valued by organisations and goverments today, includes the ones demonstrated in Figure 2.3, which are HTML, HTTP, TCP/IP, SQL and J2EE (open std.org, 2002; Shewale, 2009; Simon, 2005).

These are open standards that have been utilised by the IT industry, organisations and governments since the widespread adoption of OSS. According to Simon (2005) these particular open standards provide organisations and governments with solutions that range from interchanging vocabularies via the Web to defining strategies for business.

HTML is among many other open standards recognised by organisations, governments and the IT industry as a whole. It is used practically everywhere on

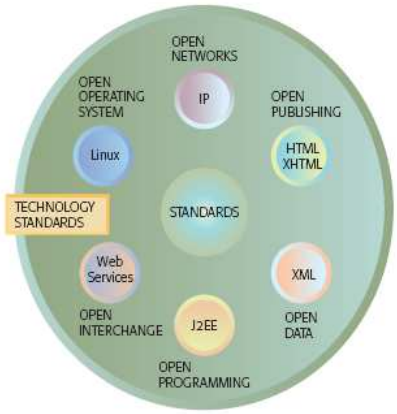

Figure 2.3: Open standards valued by organisations and governments

the World Wide Web. Using an open standard such as HTML guarantees that the IT industry, organisations and governments will no longer suffer from a lack of flexible Web services (open std.org, 2002; Simon, 2005). The Internet itself is indeed a prime example illustrating the value of open standards. It simply would not function in the absence of documented standards, such as HTML (Simon, 2005). XML is also a promising open standard utilised for seamless sharing of information among governments, reports Simon (2005). He also explains that the more accepted open standards like HTML and XML become, the more clear it will be that organisations and governments can benefit as they move toward OSS adoption (or adopt more e-government services).

2.3.4 The benefits of using open standards

As a standard used both in OSS and proprietary software, open standards have similar benefits to those of OSS. Openness, transparency, interoperability, vendor lock-in, choice, flexibility and support are the key benefits identified by Perens (2007), Simon (2005), Krechmer (2005) and Ghuha as cited by Shewale (2009), Cerri and Fuggetta (2006) for utilising open standards in organisations and governments, and are as follows:

1. **Openness.**
   Perens (2007) and Ghuha as cited by Shewale (2009) explained that open standards are open to anyone (or any interested party) that wants to participate during its development and at a final stage. 'Anyone' refers to any industry, individual, the public, government bodies or academia worldwide. According to Cerri and Fuggetta (2006) once the standard has been published it is available either freely or at a nominal charge. This means that it can be copied, distributed and used at a nominal fee, sometimes at no fee at all.

2. **Transparency.**
   Open standards are usually made public instead of being kept confidential.

This means that all their design, processes, all the technical discussions and meeting minutes are public, archived and made referenceable when it comes to decision making (Ghuha as cited by Shewale 2009). "The standards are adopted and maintained by a non-profit organisation, and its ongoing development is open for discussion even the decision-making procedure is publicly made available to all interested parties" (Cerri and Fuggetta, 2006) .

3. **Interoperability.**
   This is the term that was previously mentioned as one of OSS characteristics in Section 2.2.2 The term interoperability is defined by Dalziel (2003) as the ability of different software systems to exchange information in such a way that they can both act in equivalent ways on the information, leading to equivalent user outcomes. In practice he meant that, with interoperability users are not locked to one software system, they can substitute one standards compliant system for another standards compliant system. A good example, that highlights the importance of interoperability, are e-government services that have to work together independent of whether OSS or proprietary software is used to implement them. By using e-government services, Simon (2005) explains that citizens and enterprises can seamlessly exchange information electronically both within and beyond local boundaries.

4. **Vendor lock-in.**
   Krechmer (2005) and Simon (2005) report that governments are turning to open standards to alleviate the problems of vendor lock-in. There have been situations where government IT buyers and commercial suppliers misuse government systems when governments do not use open standards for their systems. The supplier chosen for the initial system will frequently change prices of software solutions for profit. It is inherently clear that competition will always exist among implementations of interoperable products or services no matter what the cost. Governments cannot operate on a limited budget which means they can simply not afford to pay inflated prices for these software solutions. Hence government now prefers to use open standards to store files that are not supported only by a single company or a handful of vendors. Therefore, open standards were introduced in governments to increase choice when it comes to data formats that are accessible to every citizen and enterprise within governments, reducing vendor lock-in (Simon, 2005).

5. **Choice.**
   This refers to the choice of running open standards in different environments; proprietary or OSS. Often OSS is obtained from multiple suppliers, Perens (2007) and Simon (2005) explains, and it can run on multiple hardware architectures (e.g. Linux, Microsoft Windows). The same applies to open standards; they can be used in any environment and with these kinds of combined options available it leads to freedom of choice and flexibility.

6. **Flexibility.**
   The environment in which governments are operating nowadays is changing more frequently than ever before, reports Simon (2005). Therefore, to succeed in this kind of environment, governments must be more flexible in handling events such as natural disasters or terrorist attacks which may occur, putting governments in the position of loosing data. To avoid conflicts during disasters like these, governments must start now to integrate all its information, databases and information systems. The unpredictable nature of these emergencies requires flexibility, interoperability and transparency of government systems, explains Simon (2005).

7. **Support.**
   On-going support and maintainance for open standards are provided over a long period of time, explains Ghuha as cited by Shewale (2009). This support includes among other things multiple implementations, ongoing processes for testing, errata revision and permanent access.

The next topic will address open formats, which is more aligned with open standards as they can both be implemented in proprietary and OSS environments.

2.4 OPEN FORMATS

Open formats are relevant to this research study because of the relationship with OSS. In summary Simon (2005) explains that an open format is a subset of an open standard. Taraborelli (2004) defined it as a published specification for storing digital data, usually maintained by a non-proprietary standards organisation and free of legal restrictions on use. "The primary goal of open formats is to guarantee long-term access to data without current or future uncertainty with regard to legal rights or technical specification. A common secondary goal of open formats is to enable competition, instead of allowing a vendor's control over a proprietary format to inhibit use of competing products" (Simon, 2005). As much as goverment has bought into the idea of open standards, they have increasingly shown an interest in open format issues as well (epractice.eu, 2007).

2.4.1 Open formats vs OSS

The difference or the relationship between open formats and OSS is often misunderstood, explains Simon (2005). He says many proprietary software products use open formats and OSS sometimes utilises proprietary formats. For example, "HTML, the well recognised open format markup language of the World Wide Web, can be utilised in both proprietary web browsers such as Microsoft's Internet Explorer and in an OSS browser such as Mozilla Firefox" (Simon, 2005). Other examples include OpenOffice.org, which is an office suite for desktops and can be used both in a proprietary and OSS environment, as well as Adobe's PDF (BusinessTimesOnline, 2005; opendocumentfellowship.com, 2009). However, "some proprietary formats are covered by some form of patent restrictions that may forbid OSS implementations" (Simon, 2005), except if it is under General Public License (GNU). Further examples below demonstrate some open formats as described by NETC.org (2003) and Simon (2005).

2.4.2 Examples of open formats

- ODF (for office documents and suites)
- PDF (for documents)
- LaTeX (a document markup language)
- DVI (a page description language)
- TXT (an unformatted text format)
- HTML/XHTML (a markup language)
- OpenEXR (an image format)
- PNG (a raster image format)
- SVG (a vector image format)
- VRML/X3D (realtime 3D data formats)
- FLAC (an audio format)
- OGG (audio formats & a video format)
- XML (a markup language)
- ZIP and 7Z (data compression formats)

In summary, it is essential that open standards and formats be utilised both in an OSS and proprietary environment, to ensure that the released software products conform to the outlined standards and are fit to be applied in both environments. Without open standards and formats, OSS is incomplete.

2.5 OVERVIEW: PROCESS REFERENCE MODEL STRUCTURE

After the discussion about the general importance of OSS and its adoption, as well as open standards and formats the next topic will be process reference models. The purpose of this section is to provide a theoretical overview on the usefulness of process reference models, as well as the role they play in an organisational context. By definition a process reference model (sometimes known as the generic model, knowledge model, universal model or model patterns) is informative material consisting of a library or knowledge repository of processes discovered during a certain activity within a specific environment, then classified into a process model diagram to graphically give an overview of the flow between the processes and their sub-processes (US_Homeland_Security, 2008; van der Merwe, 2005). As a set of generic process models, process reference models are used to promote reuse of processes (van der Merwe and Kotzé, 2008). The main objective of a process reference model is to assist enterprises that perform similar practices with reutilisation of proven processes, without having to develop their own from scratch (Rosa et al., 2005). Also, a process reference model reduces the risks and costs associated with repetitive errors

of the same nature that tend to happen during the operation of a particular business or project (Jensen and Scacchi, 2003). The disadvantage of using uncaptured common practices in a given domain (or enterprise) is that it may lead to errors during operation, which may come at high cost and without any guarantee that they will not take place again. Therefore documenting practices of the same nature in a given domain will provide consistent and satisfactory results for that particular enterprise and for other companies when adapted in multiple projects (Jensen and Scacchi, 2003; Rosa et al., 2005).

van der Aalst et al. (2005) reckons that process reference models are like 'plug and play', they get developed and often require further improvement to reach perfection. Addressing this issue also is Tyrrell (2000), who indicated that a working process needs to be monitored and improved from time to time to ensure that it meets the requirements and possible predictions that it was initially intended for.

Process reference models are represented in various modelling languages and standard notations (Fettke et al., 2005). Therefore a defined standard notation is required for the configuration of a process reference model and the notation in which process reference model is captured must be widely accepted and be easily understood by any individual without any modelling expertise (Rosa et al., 2005). Notations like Integration DEfinition for Function modelling (IDEF0) which is widely used for the creation of process models (FIPS_PUBS, 1993), has been selected for this study as the preferred notation to be used for modelling the generic processes captured during the CSIR's migration to OSS. IDEF0 is one of the serial set of standard reference methods for process modelling that the IDEF family of methods comprises of, created by Wisnosky and Shunk. It is a standard component that can be found and used both in Microsoft Windows and Linux for process mapping (Sun, 2006). Other commonly accepted process modelling notations used to present business processes or any other form of a process include amongst others Business Process Modelling Notation (BPMN), Enhanced Line of Visibility Enterprise Modelling (LOVEM-E) and Architecture of Integrated Information Systems (ARIS) (Brain et al., 2005), and are discussed in detail in Section 2.5.6. Other modelling languages that are used for specifying notations include Unified Modelling Language (UML) and Ericsson-Penker business extensions (Havey, 2005) and are also discussed briefly in Section 2.5.6.

Fettke et al. (2005) described a framework consisting of criteria used to address the construction of process reference models. The phases of the framework consisted of these stages, namely: the familiarisation with the application domain, identifying the process modelling languages to use during the construction of the process model, size of a model, evaluation and application of process reference models. Similarly, Schreiber and Wielinga distinguished three stages in more detail that involve the construction of a knowledge model, known as the process reference model. Also, a systematic approach that can be used by model developers to identify a process reference model structure was suggested by van der Merwe and Kotzé (2008) describing five phases that can be followed during the identification of process reference models (more details in Chapter 3). When applying these approaches, process reference model developers may become aware of the fact that modelling can be carried out in many different

ways and not just in a single optimal path because no single correct solution exists for this constructive activity. Also, support is given through a number of guidelines that exist and that have proven to work well in practice (Schreiber and Wielinga, 2005).

A number of process reference models have been proposed in publicly available sources, where some of the well-known process reference models have been identified and investigated, among others, by Fettke *et al.* (Fettke et al., 2005). For example, Supply-Chain Operations Reference-model (SCOR) was "developed and endorsed by the Supply-Chain Council as the cross-industry standard diagnostic tool for supply-chain management. SCOR enables users to address, improve, and communicate supply-chain management practices within and between all interested parties" (Stanford-Medical-Informatics, 1996). Another one is a comprehensive SAP software process reference model developed by SAP AG and IDS Scheer which covers more than 1000 business processes and inter-organisational business scenarios (Pesic and van der Aalst, 2005);(van der Merwe and Kotzé, 2008).

Jensen and Scacchi (2003) developed a process reference model for OSS community processes and but not necessarily for OSS migration. It served as a guide to different open source communities and their various OSS project artifacts to help them code and model OSS processes taking place within their communities. The aim of this process reference model was to enable these communities to succeed in terms of receiving consistent and proper contribution and also to grow. The process reference model enables the project founders and the team to clearly think of how to make the community work, be it as a new community created from scratch or an existing one that needs revival (Jensen and Scacchi, 2003). A dictionary definition of project artifacts refers to documented outputs and work products specific to a project implementation (Reference.com, 2006). Examples of common project artifacts in OSS include webpages, chat transcripts, defect reports, source repositories, development and community infrastructure tools and development resources (Jensen and Scacchi, 2003). To prove the effectiveness of the process reference model, the modelling of processes were applied to seven case studies and among the list were NetBeans IDE, Apache HTTPD server and Mozilla projects. Due to the various project's processes the resulting process models produced an overwhelming amount of data to examine. The resulting process models were then tested against other case studies of the same nature to further prove its validity. Similar processes was followed for the identification and capturing of CSIR OSS migration process models for the scope of this study.

Over the years process reference models were used by organisations to provide them with generic solutions. It is apparent that these solutions enabled different application domains to improve their business performance. Ideally process reference models often require some customisation or configuration to enhance and extend the way business operates (van der Aalst et al., 2003). From the experiences of other organisations, it is evident that process reference models are the most powerful means for transforming business performance (Childe et al., 1997). To understand the whole view of a process reference model structure, one needs to know what process modelling is all about, as well as all the concepts that are potentially related to process modelling. See the below section

for detailed discussions on process modelling concepts.

2.5.1 Process Modelling

The concept of process modelling is viewed differently by various authors, Perumpalath (2005) and van der Merwe (2005) defines process modelling as "the procedure of constructing the process model using a standard notation". At the other end of the spectrum, BusinessRanks.com (2006) refer to it as "a graphical representation of the business processes, activities, actions, and operations that capture, manipulate, store, and distribute data between a system and its environment and among components within a system", while Uvium_Inc. (2005) views it as "the use of information and graphics to represent processes in a consistent way". For the context of this study the definition provided by Perumpalath (2005) and van der Merwe (2005) for process modelling will be the one that will be used, as it can be easily comprehended by readers.

There are several purposes for which process modelling can be done, the most important one being documentation (Michael, 2009). Documenting activities of a process contributes to the success of many projects. It can be in the form of documenting already existing processes that are currently running or sometimes documenting newly emerging processes designed from scratch. It does not matter how it is done, the most important thing is to have it written down and stored in a repository. Other reasons that businesses use process modelling are: process reorganisation, process monitoring and controlling, continuous improvement, quality management (ISO 9000), benchmarking (to compare with best), practice and knowledge management (Barn, 2007).

2.5.2 Intended use of process models

According to Holland (1996), a process model on its own is a tool. That means it provides a means of communicating complex business functions in a form that will be easily understood by people. Process models are used to capture and track an organisation's practices; how it currently operates and what actually happens during a process (Childe et al., 1997). This suggests that process models enables humans to determine how the process is working somewhere else. Furthermore, process models can be used to describe the structure and process perspective of the organisation at the highest level and on the lower levels (van der Merwe, 2005). Childe et al. (1997) states that before striving to build process models, one needs to first identify what a process is and by doing so one will be able to understand the way in which a process must be constructed. Furthermore, van der Merwe et al. (2007) goes on to specify that a process must first be available before it can be used, as it can be time-wasting to work on a process that does not exist. According to Bider (2002, revised in 2003), process modelling is becoming increasingly popular and is applied on a large scale for various purposes, which are:

1. To be Descriptive, that means to:
   a) track what actually happens during a process.
   b) take a look at the way a process has been performed and determine the improvements that have to be made to make it perform more efficiently.

2. To be Prescriptive, that means to:
   a) define the desired processes and how they should be performed.
   b) lay down rules, guidelines and behavior patterns which, if followed, would lead to the desired process performance.

3. And to be Explanatory, that means to:
   a) provide explanations about the rationale of processes.
   b) explore and evaluate the several possible courses of action based on rational arguments.
   c) establish an explicit link between processes and the requirements that the model needs to fulfill.

These purposes for process modelling enabled the CSIR migration project to achieve its aims with process modelling by capturing the project's processes that was observed during the movement from a proprietary (Microsoft Windows) platform to an OSS one (Linux) and developing a process reference models structure based on the results of those captured processes.

van der Merwe et al. (2007) found that building a process model structure can be complex and costly. This problem may only be overcome by reutilising a set of generic models which may be adapted to fit a specific environment of the organisation (Childe et al., 1997). Generic process models allow humans to learn from the designs of other projects (Tyrrell, 2000), which have underwent the same route, enabling them to apply similar processes without having to build their own from scratch. Childe et al. (1997) reckons that the construction of process models is a resource intensive activity, and such resources can come in very handy when used for reference purposes. As pointed out earlier in Section 2.2.5, documentation of processes are efficient for reference purposes.

The construction of process models for the OSS migration during the study was contextualised by a five stage approach (or methodology) by van der Merwe and Kotzé (2008) called the systematic research approach for educational institutions. Among those stages is:

Phase 1 - Defining the scope (application domain or scenario)
Phase 2 - Identifying a procedure that will be used to develop the process model structure
Phase 3 - Gathering data
Phase 4 - Comparing results
Phase 5 - Verifying the obtained results.
Detailed descriptions of these phases are provided in Chapter 3 (see Section 3.2.3.1)

It is essential for process model developers to consider the idea of this hierarchy when building a generic process model structure that will benefit other application domains (or other businesses and projects). For each stage a number of techniques (or processes) exist, which can be decomposed further into subprocesses.

2.5.3 Generic process models

Generic process models are used as the intervention tool that "encourages or enables the process developer to reuse what has already been identified previously and extend only if needed" (van der Merwe et al., 2007). A generic process model usually comprises of reusable processes intended for representing the business process perspective of an organisation (Browne and O'Sullivan, 1995).

Generic process models are carried out for a number of reasons:

- to provide pointers to solution patterns and designs (Sutcliffe, 2002).
- to allow participants (or others) to learn from previous project's designs (Tyrrell, 2000).
- to provide continuous improvements that fit specific company scenarios (Tyrrell, 2000).
- to re-engineer a business / re-think the way business operates currently (Childe et al., 1997; Tyrrell, 2000).
- to meet customer expectations and changing market needs (Tyrrell, 2000).

Generic process models are constructed using notations and methodologies to facilitate the connection of activities taking place within an organisation from the high-level to the lower-level (Sotirios Koussouris, 2007). Childe et al. (1997) proposed that the application of generic process models in a specific domain can be used for process transformation.

2.5.4 Process

A process on its own is regarded as an action. Furthermore, processes are referred to a set of related actions (sometimes called tasks) performed to achieve a specific goal (Barn, 2007), (Jensen and Scacchi, 2003). There are many varying definitions stating what a process is, see below.

The term "process" as defined by others:

- Silva (2006) describes a process as a "manual or computerised activity or function that is performed for some specific business reason".
- Davenport and Short (1990) as cited by (Malhotra, 2006) defines a process as "a structured, measured set of activities designed to produce a specified output for a particular customer or market".
- A definition provided by JISC infoNet that best describes a process is that "it is a particular course of action intended to achieve a result" (JISC-InfoNet, 2007).

According to Tyrrell (2000) there are a number of significant goals that a process intends to achieve namely: effectiveness, maintainability, predictability, repeatability, quality, improvement and tracking.

1. *Effectiveness*: The more effective the process is, the better. An effective process helps to produce the right product or software. It doesnt matter how well-written it is, as long as the process determines the customer's requirements, produces and verifies what the customer needs then it is of use.

2. *Maintainability*: Every now and then a product or software requires maintenance. At times versions or customers' requirements might change. With a written process in place it will help in such cases "to expose the designers and programmers thought processes in such a way that their intention is clear. That way developers can quickly find faults and work out the solution of where to make changes" (Tyrrell, 2000).

3. *Predictability*: When developing a product it is often required to predict (or estimate) accurately the time it will take to develop the product and nothing but a useful consistent process can help to do that. Therefore, by laying out a hierarchy of steps (or useful consistent process) it will help to develop and produce a good product that will fully meet users' needs.

4. *Repeatability*: This refers to a replication of the process. Once a process is discovered to work then there is a possibility that it can be repeated in future projects, this procedure is also referred to as process reuse.

5. *Quality*: The quality of the process is as important as the quality of the product or software. The intention is to ensure that the product fits the customer's desires and it is of good quality for assurance purposes.

6. *Improvement*: Often a process must improvement further to reach perfection and reflect all possible changes that might come along.

7. *Tracking*: This means to keep track of the process, ensuring that it does meet customers' requirements. If it doesn't then it must be improved upon until it satisfies the objective of having been implemented in the first place.

Not only can these process goals be applied in a single project, they all comprises of a set of instructions describing what activities must be performed and what documents must be produced (Tyrrell, 2000).

A group of effective processes as a whole, make up a process model, modelled using a process mapping tool. Therefore it is of importance to reuse processes that have been discovered to work consistently, to ensure that these processes support the plans that were clearly stated upon the initial execution of the project. As specified earlier in Section 2.5.3, without a notation it can be a challenge to capture processes. Section 2.5.6 outlines more the different kinds of existing process notations.

2.5.5 Process Structure

As specified previously processes are event-driven (Holland, 1996), that means they comprise of an activity or event that will trigger it to produce a result. Below is process structure by Holland (1996) diagrammatically revealing the key aspects of a process.

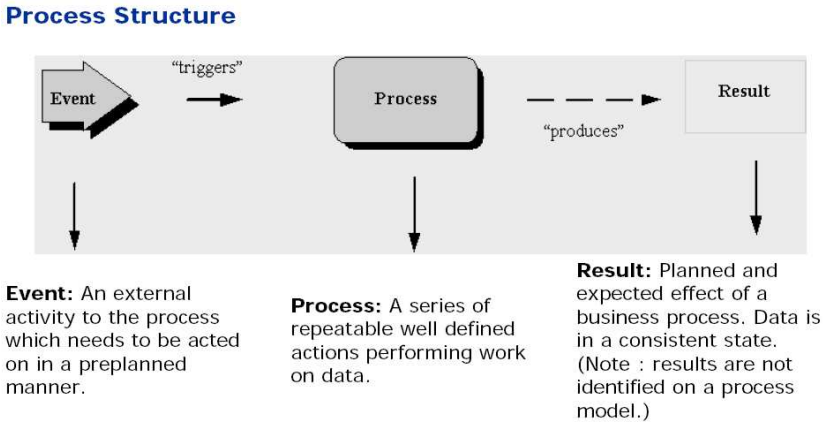

Figure 2.4: Design structure of a process

An event in this instance is the 'trigger' for a process, that is it causes work or an action to take place (Holland, 1996). Sometimes the term event could refer to the execution of data or update of data as per organisation's process. If there is no event conducted, then it means no work was done. Holland (1996) further suggested that each event on a process model must decompose into other processes or elementary processes; without a corresponding elementary process major events will make up a process model. The concept discussed is simple: without an event it means no work was done (no elementary process equals no event or work done). In a few cases, events will comprise of an elementary process and in such cases it means that no work was done to produce results.

2.5.6 Process notation

According to Havey (2005) one way to present a process, be it for a business or project, is by making use of notations. One of the most important parts of the process modelling phase is choosing an appropriate modelling notation. Because the selection of a right notation can increase chances of producing potential generic process models (Bider, 2002, revised in 2003) and may also be "critical to the success of the process analysis to be performed, if all the elements that influence process development are represented by the modelling notation" (Brain et al., 2005). According to van der Merwe (2005), these elements of the modelling notation includes "symbols used in the models and rules that govern the use of the symbols".

Dozens of notations exist, some have been designed to suit specific needs or company scenarios. As pointed out the most popular types of notations that are

commonly used to model processes include, IDEF, BPMN and the UML. Furthermore, there is also Ericsson-Penker business modelling profile extensions, which are notational languages that includes amongst others UML, BPMN, Ericsson-Penker business extensions, BPEL and ER. A bit of each notational language will be discussed in this section.

2.3.5.1 Notational languages

IDEF0 originally created by Wisnosky and Shunk as part of the IDEF family of methods for process modelling. It is a standard component that represents Function Modelling and can be found and used both in Microsoft Windows and Linux for process mapping (Sun, 2006). It is also a standard notation widely used for the creation of process models and was selected as the preferred notation for this study to model OSS migration processes for the CSIR project.

UML it is simply a graphical modelling language (de Paula, 2006), used for visualising, specifying, constructing, and documenting the artifacts of a software intensive system (SparxSystems, 2007). It is associated more with software engineering and systems' design than with analysis and modelling of business processes. As said UML is a 'language', therefore it has a very particular characteristic with diagrams as its notation as reported by de Paula (2006). Furthermore it is used for specifying, but in itself it does not specify the methodology or process or procedure.

BPEL formerly known as Business Process Execution Language for Web Services (BPEL4WS8), defines a notation for specifying business process behavior based on Web Services. It is used for modelling the behavior of both executable and abstract processes. That means modelling of process activities, especially of Web Service interactions, Web Service based relationships between process roles and correlation of messages and process instances (Barry, 2000).

BPMN its a standard workflow notation that is easily understood by all business users. It can be used with BPEL and other business process languages. Over the last couple of years BPMN became the most popular process modelling technique used by many businesses and organisations motivating them to make significant investments in process modelling initiatives (Rosemann et al., 2007). This statement implies that BPMN is the popular process modelling technique used by organisations for their process modelling activities.

ER is a modelling notation originally proposed by Peter Chen in 1976. As Chen's original notation, ER uses rectangles for entity types, diamonds for relationships, and ellipses for attributes. It views the application domain in terms of entities that have attributes and participate in relationships. In spite of the recent rise of a UML, ER is still the most popular data modelling approach for database applications (Halpin, 2006).

Similar to BPMN, **Ericsson-Penker profile** is a UML extension that exists to further enhance the capturing of business processes (SparxSystems, 2007). The extensions for this profile are simply called the Eriksson-Penker Business Ex-

tensions and they provide symbols for modelling the processes, resources, rules, and goals of a business system (Eriksson and Penker, 2000). The extensions merge UML with process modelling, so that it is easier to use UML for business modelling. Anyone familiar with UML should be comfortable with these extensions, as will those who are knowledgeable in business process modelling.

Brain et al. (2005) have identified notation requirements which were established by reviewing an extensive amount of the existing literature.

2.6 MOTIVATION FOR THE RESEARCH

Dudley et al. (2006), van Reijswoud and Topi (2003), Simon (2005), Cerri and Fuggetta (2006), Hoe (2007), Opensourceafrica.org (2007) and GoOpenSourceTaskTeam (2003) reported on the importance of adopting OSS within organisations and government departments worldwide. While Bulusu (2003), open std.org (2002), Fremantle (2003), NETC.org (2003), Taraborelli (2004), Krechmer (2005), Perens (2007) and others emphasised on the usefulness of open standards and open formats in relation to OSS.

This study first focussed on providing theory about OSS and process reference models in Section 2.2 and 2.5. The background on the CSIR as the environment of study, as well as all concepts known about OSS migration, open standards, open formats, process, process models and generic process models (process reference models) were discussed. Therefore, the aim for conducting this research study is to capture OSS migration processes that can be reused by other organisations and companies embarking on the same type of a project in future, without having to develop their own processes from scratch. This research study will provide a set of process reference models comprising of reusable migration processes.

2.7 SUMMARY

This chapter described the theory behind OSS and process reference models in Section 2.2 and 2.5 in relation to the first two research questions, as well as the reuse of processes mapped using a process model tool as a generally accepted representation of a process reference model structure. In terms of the first research question, different concepts of the process reference model structure were discussed which included the process, process modelling, process structure, generic process models and the notation. As for the second research question the OSS concept was discussed from a business perspective to a migration perspective. The relationship between open standards and open formats with OSS was investigated to provide a brief definition of each concept. The other two research questions will be addressed in Chapter 4 and 5 as part of the identification of OOSM activities. The systematic research approach by van der Merwe and Kotzé (2008) consisting of five phases for identifying a process reference model structure was suggested in Section 2.5.2, but only three of its five phases will be utilised for this research study.

Chapter 3

RESEARCH DESIGN AND METHODOLOGY

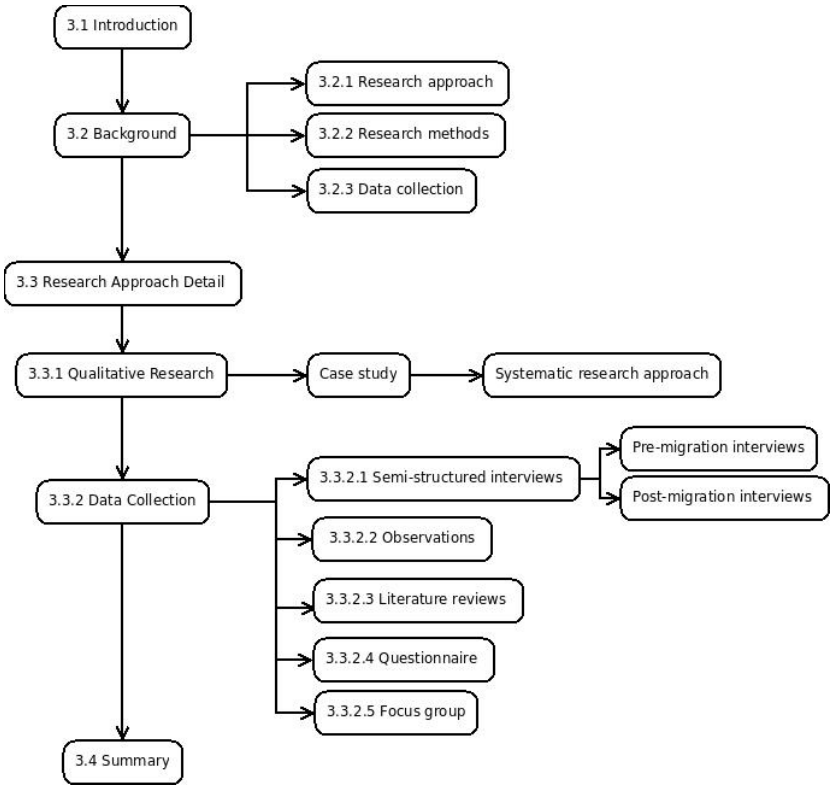

Figure 3.1: Chapter 3 Outline

3.1 INTRODUCTION

The purpose of this chapter is to provide background on the research design used during the study and to describe the research methodologies that assisted with reaching the intended results for this specific research study. The main objective of this study was to extract process reference models for an OSS migration project from a set of process models modelled using the process modelling notation. The research design is a platform that defines what information must be gathered, who from, how and at what stage (USEPA, 2007). The data gathering methods (or instruments) that will be used must be identified and then then data obtained must be organised and analysed.

Section 3.2 provides the theoretical overview of the research approach used during the study, followed by the description of research methods that were used to explore research questions and the data collection tools used for gathering data. In Section 3.3 the theory on the research design approach used in this study is applied. The chapter concludes with the research methodologies and design approach and data collection techniques used during the study. The contents of this chapter are outlined in Figure 3.1 above.

3.2 BACKGROUND

This section addresses different research approaches and methodologies that could be applicable to the study (see Section 3.2.1 - 3.2.3)

3.2.1 Research Approach

In research, there are several types of research approaches. One approach is the qualitative study that is referred to by Myers (2004) and Moore et al. (2007) as a study that involves the collection and interpretation of qualitative or non-numerical data. It uses interviews, observation and document review to collect data (USEPA, 2007). These characteristics of qualitative research assist the researcher to explore and understand people's views, attitudes and experiences about a specific situation (Myers, 2004). In contrast to qualitative research, there is quantitative research which is about measuring the quantity of things and collecting and generating numerical data (Neill, 2007).

Qualitative study can be time consuming according to Neill (2007), but has the tedency to be richer and less prone to generalisation because it is meant to provide complete and detailed descriptions of data. Qualitative research uses data collection methods to gather data. These data collection methods are sometimes called research methods. These methods namely, *case studies, ethnographic, action and interpretive research, critical social theory,* and *grounded theory* (Myers, 2004; van der Merwe, 2005) are mostly used in the information systems (IS) field and are described briefly in the next section.

3.2.2 Research methods

There are a number of qualitative research methods available. Vaishnavi and Kuechler, as cited by Bechan (2008) described a research method as the set

of activities a research community considers appropriate in order to gain understanding of the subject or to produce new knowledge. He went further by saying a research method is often required within qualitative research to assist with the progress of doing the actual research (Bechan, 2008). Therefore, when choosing the research method to use for a specific qualitative study, select the one that will impact the way in which data will be collected by the researcher. Qualitative research methods are designed to assist researchers in understanding people and explain the social phenomena within which they live (Myers, 2004)

In Section 3.2.2.1 until 3.2.2.6 a brief overview is provided of some of the popular qualitative research methods, while Section 3.2.3.1 describes the method suggested by van der Merwe and Kotzé (2008) utilised for gathering data during the study.

3.2.2.1 Ethnographic research

Ethnographic research is one of the most in-depth social science research methods and is commonly used by anthropologists to study cultures and societies (NCREL, 2008). According to Myers (1999) "all types of ethnographic research require an ethnographer to spend a significant amount of time in the field". To perform research within the field relies on personal experiences, possible participation and observation by researchers or ethnographers (Genzuk, 2003). Like any other research method, ethnography has its benefits and one of the main important benefits of ethnographic research is the sense that an ethnographer or researcher is always present during an event to observe everything that people are doing and saying for a significant period of time (Myers, 1999). That way, an ethnographer will gain a complete understanding of the scenario, the people, their behaviours, routines and the dilemmas of their everyday life that may challenge assumptions.

Ethnography is commonly used in the study of Information Systems in organisations from, for example, the study of the development of information systems to the study of aspects of information technology management (Myers, 2004).

Ethnographic research was not used for this study because it does not investigate, it only requires the researcher to spend a meaningful amount of time engaging in the lives of people they are studying (Smuts, 2008).

3.2.2.2 Action research

Action research is a particular approach of research that is qualitative, interpretive, reflective and experimental and also uses ethnographic techniques such as: field notes, participant-observation, interviews, audiotaping and collection and analysis of data (Myers, 2004). It is defined by a student from Kings_College_London (2009) as research that occurs when a researcher designs and conducts an experiment in a field, collects data and then gives it back to the participants for feedback and as a way for modelling the next phase of the experiment (Kings_College_London, 2009).

Action research is commonly applied as a research method in fields such as organisational development and education, except that in Information Systems this research method has been largely ignored for a long time (Myers, 2004).

Action research was not used for this study because it combines theory and practice, that is, the researchers and the practitioners with mutual goals. It fulfils the needs of the study subjects as well as new knowledge. It applies theory with the goal to enhance (Myers and Avison 2002, Kock, Avison, Baskerville and Myers *et al.* 1999), as cited by Smuts (2008) and van der Merwe (2005).

3.2.2.3 Interpretive research

Interpretive research involves the exploration of human behaviour, to enable researchers to gain a better understanding of the processes that may influence the behaviour (Kings_College_London, 2009). It also allows them to interpret any observed phenomena of humans through meanings, languages and other artefacts (Berntsen et al., 2004).

Interpretive research is often described as qualitative, as it usually involves using qualitative methods from which to understand the data that was collected and analysed during the research process (Berntsen et al., 2004). However, the research is not necessarily interpretive just because the type of data collected is of a qualitative nature, because there are other ways of using numerical data in interpretive research. Interpretive research originates from the social sciences and is widely accepted in Information Systems research.

According to Klein and Myers (1999), cited by Berntsen et al. (2004) interpretive research aims to seek meaning between a social and historical context to enable an understanding of how the current situation emerged. The research is assumed to be interpretive only if the knowledge gained was through social constructions such as language, consciousness, shared meanings, documents and other artefacts (Berntsen et al., 2004).

There are some elements of interpretive research in this study, although it was not used as the main approach. Some elements are similar to those of a case study. For instance, interpretive research uses social constructions such as documents and shared meanings to gather data that will allow the researcher to gain knowledge of the study, while a case study uses documentary evidence for data collection (see 3.2.2.6).

3.2.2.4 Critical social theory

Critical social theory as defined by Leonardo (2004) "is a multidisciplinary framework with the implicit goal of advancing the emancipatory function of knowledge". A critical social theory facilitates by highlighting the relationship between social systems and people, how they produce each other and then it strives to finds the connection between the two (Leonardo, 2004). With critical social theory, the researcher is mainly involved with social activities (van der Merwe, 2005). Through critical social theory, quality is the key.

This method was not used because it covers mainly social activities or interactions between explanatory, normative and ideological dimensions of social and political thought (Centre for Critical Social Theory, 2002) cited by Smuts (2008).

3.2.2.5 Grounded theory

Grounded theory is a qualitative research methodology that generates theory from the data (Borgatti, 2001). The resulting theory can be generalised to fit the data. Such theories emerge from observing a group's experiences and allow researchers to add their own insights into why those experiences exist (CSU, 1993). Grounded theory approaches are widely used in Information Systems research literature because the method is valuable in developing context-based, process-oriented descriptions and explanations of incidents (Myers, 2004).

This method was not used because it develops theory from data that has been systematically gathered, grounded and analysed (Myers, 1997, Olivier, 1997) cited by Smuts (2008).

3.2.2.6 Case study

Holetzky (2003) defines a case study as one of several ways of doing research which involves a study of a person, situation or a specific case. Other well-known case study researchers like Robert K. Yin, Robert E. Stake and Helen Simons have proposed six steps that can be utilised when conducting case study research (Soy, 1997). For instance,

1. Determine and define the research questions
2. Select the cases and determine data gathering and analysis techniques
3. Prepare to collect the data
4. Collect data in the field
5. Evaluate and analyse the data
6. Prepare the report.

Only step 4 was utilised for this study to gather data.

There are three examples of case studies, according to Yin (2003), explanatory, descriptive and exploratory:

- An explanatory case study is used in instances where the reseacher may have little or no control over the phenomenon (or event). The case study answers questions about how the phenomenon works and why it works and enables the resarcher to draw conclusions from different theories.

- A descriptive case study is used to provide a rich, thick and detailed description or analysis of a phenomenon and its context and assists in telling a story about the different perceptions of the phenomenon's context and less detail about its mechanisms.
- An exploratory case study facilitates in defining the questions or hypotheses for the study, to help understand a problem investigated.

The case study method was considered for data gathering, together with the systematic research approach discussed in Section 3.2.3 below.

3.2.3 Research Design

3.2.3.1 Systematic research approach (van der Merwe and Kotzé, 2008)

This is a research approach suggested by van der Merwe and Kotzé (2008) for identifying and capturing process model structures, consisting of five phases as depicted in Figure 3.2.3 below. This is the approach that can be followed by process model developers during the identification of process models.

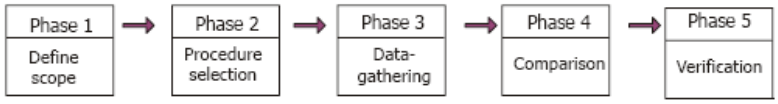

Figure 3.2: A systematic research approach [adapted from van der Merwe and Kotzé (2008)]

This approach, as shown in figure 3.2.3, has one thing in common with a case study method, that is they both have a *data collection or gathering step* (see step 4 above in section 3.2.2.6 and phase 3). The manner in which this step was used together with all other phases of a systematic research approach is discussed in section 3.3.1.

The phases of this approach were initially used for the identification and capturing of process models in different educational institutions and are briefly summarised in the following section.

In phase 1, the *definition of scope*, the development team or process model developer is required to define role players and the key personnel that will be responsible for the team. The field of the study (or working environment) that will be used for data collection must also be included. Then a feasibility study with regard to time mangament, human resources and financial implications must be conducted (van der Merwe and Kotzé, 2008). This is necessary because the top management has to decide if a project is worth pursuing and must approve it.

In phase 2 with the *identification of a procedure to derive the process model structure*, the team must select a procedure they will follow to do requirements elicitation that will help them derive the process model structure. The goal of a requirements elicitation procedure is to derive the process model structure from the identified and captured processes. It is not always the case that the team must derive a new procedure from scratch, they may have the choice of selecting existing procedures. There is no doubt that using existing procedures may save them time but if there isn't any procedure that meet their requirements then the team may consider the possibility of having to create one from scratch, using a standard process model notation and widely accepted guidelines for the construction of the process model structure. However, the team must select a procedure that can be represented diagrammatically in a model with a flow of processes (van der Merwe and Kotzé, 2008).

The *data-gathering* phase involves the selection of data collection techniques that are appropriate for conducting the activity. The selected procedure from Phase 2 must be considered to help derive different process model structures for any other specific domain. The goal of this phase is to distinguish or identify the different processes within the organisations that can be captured to derive a process model structure. However it is best to choose a data gathering tool that is financially feasible and the data or any other material collected during this activity must be documented and kept safe (van der Merwe and Kotzé, 2008).

Once the different process model structures have been derived in Phase 3, for the fourth phase, *comparison*, the goal is to identify a generic set of process models referred to as the process reference models. To perform this, a process model must consist of 10 to 15 sets of processes to complete one single function and must be utilised to fit this framework of extracting process reference models from process models. Thereafter comparison tables are used to compare the results and identify generic processes on different levels (from a higher-level to lower-level).

The final phase of *verification*, exercises the results obtained in Phase 4. It confirms whether the generic process models (or process reference models) are a representation of what had happened during the process and verifies if they can work when applied in another field of the same nature. However the results have to be practised in a domain that was not used in Phase 3, that is during the data gathering phase (van der Merwe and Kotzé, 2008).

3.2.4 Data Collection

Each of the research methods addressed in Section 3.2.2.1 - 3.2.2.6 uses different techniques for collecting data. Data according to Tustin (2006), as cited by Smuts (2008), can be distinguished as primary and secondary data. Primary data is data that is observed or collected directly from people, organisations or research participants' experiences. Secondary data is the type of readable material that has been previously published, archived and can be acquired from data libraries, books, journals and articles (BusinessDictionary.com, 2007). More information about the primary and secondary data collection sources of data

for this research is covered below. These sources of data are the six sources of evidence specified by Yin (2003) and Stake (1995) cited by Tellis (1997), Collins-Brown (2006) and Bechan (2008); namely interviews, documents, direct observation, participant observation, focus group, archival records and physical artifacts.

3.2.4.1 Interviews

Interviews are a source of primary data that can be used by the researcher or interviewer to seek data directly from a participant or interviewee by asking questions. Various types of interviews exist in a research field context. The three types of interviews are identified as structured, semi-structured and unstructured.

1. **Structured interviews** are used for a wide number of reasons and ensure that each participant is presented with a specific set of questions in the same order. All paticipants must be asked the same questions in the same way, as this makes it easier for the researcher to standardise the interview (sociology.org, 2007). These interviews require a substantial amount of pre-planning and are conducted in a controlled manner allowing the researcher to explain things that may be confusing for the participant to understand (sociology.org, 2007). Structured interviews can be time consuming if the sample group used is very large, but it is a reliable source that can be used for contacting large numbers of people quickly, easily and efficiently (alnresearch.org, 2003); (sociology.org, 2007).

2. The **semi-structured interviews** are more flexible than structured interviews because the interview process is conducted fairly and openly without requiring an interviewer to ask questions in a specific order. However, the interviewer has to use an interview guide with a specific set of questions organised according to a carefully selected topic. This guide must be well thought out and prepared by the interviewer in advance to enable both the interviewer and the interviewee to give and receive information in a focused, conversational and two-way communication manner (fao.org, 1990). Due to the responses of the participant or interviewee, new questions can be brought up during a semi-structured interview.

3. An **unstructured interview** is referred to as a spontaneous and non-threatening conversation between the researcher and the participant, allowing the researcher to ask the participant a set of key questions which the researcher had formulated in advance
(BusinessDictionary.com, 2007). For an unstructured interview questions do not need to be asked specifically in an orderly manner, they can progress based on a participant's responses. Unstructured interviews are used to acquire deep knowledge and authenticity about people's life experiences (Economic and , ESDS). Therefore Herman and Bentley (1993) say that there is no one right way to do an unstructured interview; they often

contain open-ended questions which means discussions may go in any direction. The only important thing when conducting an unstructured interview is to gather and record the information as evidence during the interview process. These types of interviews are suitable for allowing the reseacher and participants to ask and answer questions, share feelings, additional ideas, thoughts and comments.

Several rules designed for conducting unstructured interviews are given below, from Herman and Bentley (1993):

(a) Avoid leading questions

(b) Probe beyond the expected answer

(c) Explore inconsistencies

(d) Record participants' own words.

3.2.4.2 Observations

According to Hammersley (2008), "observation at minimum involves a researcher watching and listening to actions and events within some context over some period of time and making a record of what has been witnessed". From a research perspective there are two types of observation which are participant and non-participant. With participant observation the observer takes part in a situation being studied rather than simply being there and not participating in a situation like the non-participant observer (in eLearning UCEL, 2004). Hammersley (2008) further said that observation, be it participant or non-participant is an alternative method to the use of interview, documentary or questionnaire data. That is, it takes place in natural settings other than in laboratories where experiments are conducted.

3.2.4.3 Literature review

As the secondary data collection technique, a literature review is defined by the Public_Health_Agency_of_Canada (2005) as the study of existing publications, documents or articles that is within the scope of the specific field of the subject. According to Elvis and Mathur (2008) and the F_D_Bluford_Library (2005), literature review is used for several purposes: (1) to illustrate the importance of the topic, (2) to convey the knowledge and ideas established on a topic to the reader, (3) to keep the reader up to speed regarding the state of research in the field, and (4) to show the reader the lack of research on the topic. Literature review keeps one up to date with the theoretical material.

3.2.4.4 Questionnaire

A questionnaire is a research instrument used both in qualitative and quantitative surveys to capture or generate data required for accomplishing the objectives of a study (NMSU, 2008). It consists of a group of questions that must be answered by respondents prepared in a computer or printed form (NADbank.com, 2008). Sometimes this set of questions are in the form of open-ended questions and close-ended questions (analytictech.com, 1997). According to CSU (1993), by definition open-ended questions require the respondent to answer in his or her own words and to even provide an explanation for their answer. Close-ended questions limits respondents' answers as it only requires the respondent to answer a question as "yes/no" or "true/false".

3.2.4.5 Focus group

A focus group which is part of qualitative research methods, is defined by AALBC.com (2007), JRS-Consulting (2007) and Kings_College_London (2009) as a qualitative research method consisting of a small number of people, brought together to provide feedback, information, opinions and feelings on the issue being discussed under the guidance of a facilitator. These people are chosen based on their ability to provide specialised knowlegde or insight to the issue being discussed (University_of_Texas, 2007).

3.3 RESEARCH APPROACH DETAIL

This section provides detail of the research approach, methodology and design used in this study. Therefore, the theory discussed in Section 3.2 about the research approach, research methods and data collection techniques will now be addressed in this section, to show how it was applied to this research study.

3.3.1 Research methods and design applied in this study

This study leans towards qualitative research, because it is based more on suggestions than exact numbers (statistics or samples) to enable the researcher to investigate the process models of an OSS migration project.A case study approach which is a qualitative research method was used for collecting data. As pointed out in section 3.2.2.6 a case study is referred to as a study of an event or specific case (Holetzky, 2003). A case study approach was suitable for this study because the design and methods are well suited for this research (namely observation, interviews and documentary evidence). These methods gives the case study approach the strength and the ability to answer the research questions addressed in this study appropriately (Collins-Brown, 2006). These methods of the case study approach are also fit for investigating the whole implementation of the CSIR migration project to OSS from its planning to completion. This

includes observing the migration right from the working environment, interviewing project managers, project team members and users who were involved in the migration, and reviewing in detail all the documentation that had to do with the CSIR migration and other OSS migration projects.

In order to confirm the envisioned research approach for this study, a systematic approach suggested by van der Merwe and Kotzé (2008) as shown in Figure 3.2.3 in Section 3.2.3.1 was used to identify a set of process models within the OSS desktop migration process. The systematic approach consists of five phases which defines the criteria for identifying and capturing process models (van der Merwe and Kotzé, 2008), that may help process model developers to decide how to go about identifying a set of process models (or process model structure) for a given domain.

Using the phases (or guidelines) from the systematic approach by van der Merwe and Kotzé (2008) and a case study approach for this study, the following objectives were addressed by this study with regards to the CSIR migration project to OSS as summarised in Table 3.1. However it is important to note that since the scope of this study is only based on a case study approach, only phase 1, 2 and 3 from the systematic research approach by van der Merwe and Kotzé (2008) were utilised. Step 4 and 5 were omitted due to time constraints, but can be covered in future research.

Phase	**Description**
Phase 1: Define Scope	The objective of this study is to suggest process reference models for a typical OSS migration project that will constitute the planning of future OSS migration projects. The CSIR executives and top management made this phase possible by approving the project at first and migrating the maximum number of the organisations desktops to OSS.
Phase 2: Procedure Selection	OSS migration projects were not performed as frequently before as they are now, thus none of the procedures (or documented set of process models) existed for this type of migrations. Therefore the procedure utilised for the CSIR migration to OSS was to identify and capture process models through the usage of a standard process model notation called IDEF0 to model the processes.
Phase 3: Data-gathering	Literature review, interviews and questionnaires were used as data collection tools for this specific study.
Phase 4: Comparison	The set of process models were compared against each other to extract process reference models which will hopefully assist project managers and teams in planning these projects better in future.
Phase 5:	

Verification	This phase tests the extracted process reference models against any other organisation planning to adopt OSS in future, to verify the effectiveness of the obtained results of the CSIR migration to OSS. The chosen organisation must not be the one that was used during data collection.

Table 3.1: Identification of process models phases

3.3.2 Data collection and analysis techniques applied in the study

3.3.2.1 Interviews

During the migration project an in-depth analysis of the CSIR was conducted, to investigate the critical issues that will enable the success or failure of the OSS migration project. Interviews acted as the primary tool for gathering information and as part of the process assessment during and after the migration.

Considering the background about interviews in Section 3.2.4.1, unstructured interviews were appropriate for this study because the participants employed by the CSIR were available mostly during working hours to provide feedback about the migration. The interviews were divided into two categories, the first one is that of pre-migration interviews which were conducted during the planning of the migration. The other category is the post-migration interviews which were used to investigate the experiences and views of migrated users.

Participants who were selected for pre-migration interviews were members of the Vula team. They were selected by the researcher to enable her to understand the process involved in planning and implementing the migration to OSS at the CSIR. Participants were initially given questions before the interview session, to enable them to provide useful data and suggestions during the interview process. The interviews were recorded and thereafter the data was gathered, analysed and reported using the approaches recommended.

Participants for the post-migration interviews were recruited by the Vula project manager, who gave the researcher a list of people to interview. At this point every participant stated how they were supported during and after the migration. This included their views, feelings and experiences about the migration. Both pre-migration and post-migration interviews were conducted as unstructured interviews to understand the project team's viewpoints and the viewpoints of the users (or CSIR employees) about the migration.

The mode of communication chosen for the interview process was to setup appointments with the participants via phone and email. Thereafter a one

on one (or face to face) interview was conducted with the participants. There was one request made by two participants to be interviewed both at the same time instead of in a one on one session. Each participant was asked questions which were designed by the researcher and analysed by the Vula project manager to tell their case story (or share their experiences) about the migration. Their responses often resulted in complaints, feelings and additional ideas about how the project team should go about handling the migration in future. Their thoughts and comments were incorporated into the final migration stage and were handled by the project manager and the team.

Unstructured interviews were appropriate for this study for several reasons explored above and in section 3.2.4.1. By briefly describing each type of interview it was easier to illustrate why unstructured interviews were considered for this study.

3.3.2.1.1 Interview process

Designing and validation of interview questions

As pointed out in Section 3.3.2.1, interviews were used as the primary source of data collection in this study. Two individuals, the researcher and the Vula project manager worked through the first draft of the interview questions which were initially compiled by the researcher. Thereafter the Vula project manager with his experience in OSS projects examined the first draft of interview questions planned by the researcher. His feedback was then incorporated with the ultimate draft of the interview questions. The final draft of the questions was sent to the rest of the project team via email for feedback. The questions were used for capturing information from the people who were involved in the migration to OSS at the CSIR. The participants were chosen by the Vula project manager to be part of the interview sessions and also agreed to participate in the research study.

A copy of the interview questions used in this research study is provided on CD (see Appendix D). The questions were organised in this order:

1. First questions focused on pre-migration activities,
2. Post migration questions followed and within this process the recognition of user experiences were also part of the questions.

The final draft consisted of 14 questions with subquestions included and it focused on capturing the user's views, attitudes and experiences about the migration. These 14 questions sought to help the project managers and the team to learn from their mistakes and improve their strategy on how to go about planning, project managing and preparing for the migration project before the distribution of the OSS desktop (in this case the Vula desktop) to the rest of the CSIR. These 14 questions mainly focussed on the views and experiences of users before and after the migration, their reasons for migrating to OSS, the outcomes from the migration, the benefits gained, obstacles encountered and

lessons learnt.

The first part of the 14 questions focused on investigating whether the interactions between the project team and users existed, the path the migration followed and whether the whole process of the migration was fully explained to users before they were migrated to the OSS desktop. These questions' focus was to learn whether users were satisfied or dissatisfied with the proposed strategy, planning, project organisation and project management of the CSIR migration project (Krishnan and Ulrich, 2001) cited by Ahmed (2005).

The second part of the questions together with three subquestions allowed users to raise their concerns or even give praises about the material (or documentation) provided during training sessions and presentations. The training sessions were conducted to prepare users for the migration and were organised in this order: (1) first an e-mail was sent to users who were categorised as ready to migrate by the project team, (2) the categorised users were then provided with various dates to select 2 days from, based on their availability, (3) after choosing the dates, users attended the 2 day training.

The third part of the interview questions included a question that asked if the participant would encourage the rest of the organisation to migrate after having experienced the migration and its consequences. This question sought to learn of the user's satisfaction or dissatisfaction with the migration process.

The fourth part included questions together with subquestions that focused on the level of productivity after the migration. These questions looked at the factors that disrupted users from doing their work when using the Vula desktop. The aim was to learn if the migration increased or decreased users' productivity when using the machine loaded with the Vula desktop.

The fifth and the sixth part of the interview comprised of questions that allowed the user to explain what they liked and disliked about the migration and to give suggestions about what could be done to make the migration better.

The seventh part consisted of a question that presented the user with a choice between the current Vula desktop and the Microsoft Windows desktop. They were asked how often they felt like going back to the Microsoft Windows desktop after they had migrated.

The final part of the interview questions focussed on learning if the user would participate in another possible migration project that will be performed by the CSIR in future.

All responses to the interviews were used as the main source of data for this research study.

3.3.2.2 Observations

Observations were used as a viable tool for gaining knowledge about the processes of the CSIR migration to OSS. They involved attendance of weekly meetings on Tuesdays and Thursdays held by the Vula project team to discuss the planning and implementation of the migration. Although observations were time consuming, these meetings enabled the researcher to have an understanding about the migration and decisions that were undertaken concerning the resources or infrastructure of the organisation, such as the skills and expertise needed to manage and support the migration.

3.3.2.3 Literature review

In terms of literature review, the data collection was conducted at two points in the study. The first data collection involved reviewing the previous researches, documents or publications and migration guides that had to do with general and organisational OSS migrations. The second occurred when identifying and evaluating the work that had already been written about the plans of the CSIR migration and the processes that had already been carried out successfully by the project team (these documents can be viewed in Appendix A). Both activities involved reading and data analysis.

For the first collection, in order to identify the number of OSS projects that had already taken place, a web search was conducted to identify previously published literature and new releases of documents about migrations from proprietary software to OSS. Figure 3.3 outlines the search engines and keywords which were used to find existing literature on these migration projects.

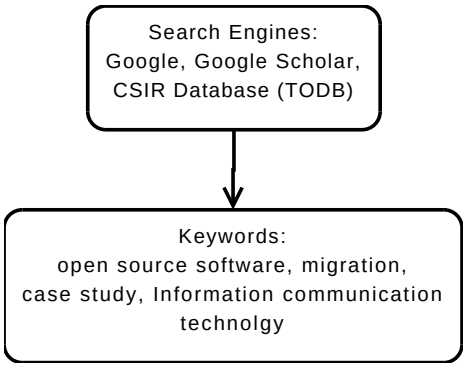

Figure 3.3: Keywords for identifying already undertaken OSS migrations

From a research perspective all literature was examined to:

- Assess whether the migration was indeed undertaken from proprietary software to OSS.
- Assess whether the migration was successful or not and if not what were the consequences.
- Name and identify people who were involved in the migration project.

This search for articles on migrations from proprietary software to OSS turned out positive with results showing that OSS migrations are currently performed popularily worldwide (see the list of links or websites on OSS migrations in Appendix B).

3.3.2.4 Questionnaire

First a Vula project website (https://vula.csir.co.za) was created to inform users (that is employees of the CSIR) of the main reasons for migrating to an OSS environment. The website was also built to allow everyone within the organisation to interact and raise their concerns or comment about the migration project. The project name, project team, those who led the migration project and championed the migration at that moment were introduced to the organisation. From time to time users were kept informed via e-mails and through the website by the project leader and executives (VulaWiki, 2006a).

In January 2006 a questionnaire was compiled by the Vula project team to assess the readiness of the organisation for the adoption of OSS and to identify the needs of users across the organisation (VulaWiki, 2006a). The questionnaire which can be accessed via this link *https://vula.csir.co.za/quests/introduction.php* was divided into six parts. In the first part the user needed to fill in their contact information. The second part dealt with Microsoft Office products. The project team wanted to find out what advanced features people use within Microsoft Office products which were or were not compatible with the OSS. The third part asked users about the proprietary software applications they were using at the time and users were even allowed to add any proprietary applications that they were using that were not appearing on the list of CSIR standard applications. Part four required the user to add any OSS (e.g. a Linux distribution) that they were using at the time. The fifth part required the user to inform the project team about their data storage; that is where do they store their CSIR data or work documents. The final part allowed the user to give comments and raise any concerns or issues that may influence them to not migrating to OSS. Appendix B provides copies of the questionnaire used for assessment during this research, but due to space limitations these questionnaire copies will be provided on CD.

Each individual within the CSIR was contacted via e-mail and requested to complete the questionnaire in January and February 2006. The questionnaire was done online, therefore the responses were viewed by the project team electronically. In some cases respondents were asked to further clarify their replies so that they were clearly understood by the project team. The responses of these questionnaires were then captured and analysed, contributing to the collection of data for this research.

3.3.2.5 Focus group

This method was advantageously used for this study to validate the collected data captured during the research. It was used to assess the validity of identified and captured process models of the CSIR migration to OSS. It is from

these process models that a set of process reference models were extracted as a representation of a generic process model structure that can can be used in future OSS migration projects.

The validation of data was performed by several Vula project team members and two study supervisors acting as facilitators. First an e-mail was sent to several members of the Vula project team with expertise in the OSS migration field, who agreed on a specific date and time to meet. The date for all participants to meet was set on the Groupwise calendar for participants to accept. The objective here was to look at the identified and captured OSS migration processes and ensure that it really was a representation of what happened in the CSIR's OSS migration project.

3.4 SUMMARY

The research methodology and design were the main focus of this chapter, whereby an approach was applied to enable the process of collecting data, that is OSS migration process. Therefore the research and methodology which were of a qualitative nature were proposed as useful instruments applicable for gathering data during the study. Qualitative research methods such as case studies, ethnographic research, action research, interpretive research, critical social theory, grounded theory and focus groups were also discussed and characterised as commonly used methodologies.

The research design approach used for this study was a systematic research approach described by van der Merwe and Kotzé (2008), and this approach was envisaged to address or answer the research questions by means of data collection techniques applied during the study.

The data collection techniques used included interviews, whereby semi-structured interviews were depicted as the appropriate primary data collection technique. Other techniques were observations, literature reviews and questionnaires. The data extracted from the interviews, literature reviews and questionnaires provided lessons and knowledge in terms of the processeses that took place during the migration project, as well as the challenges and successes that contribute towards an OSS migration project. The data was analysed using a focus group discussion.

Chapter 4

DATA EVALUATION AND ANALYSIS

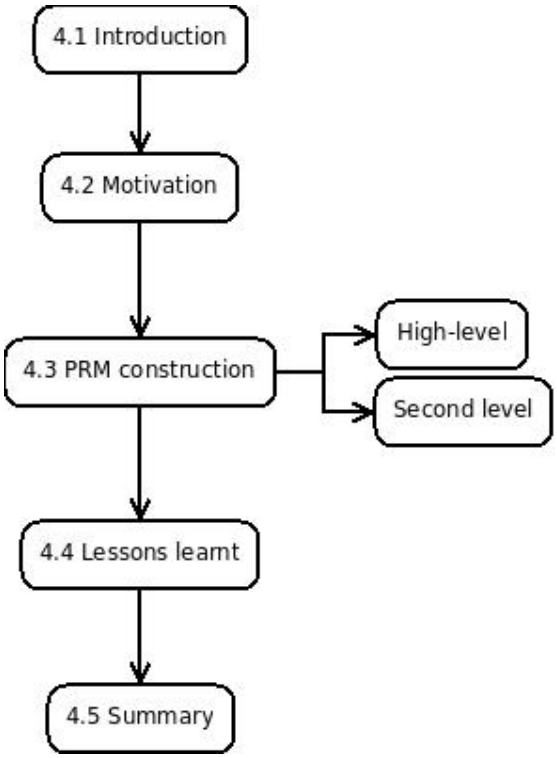

Figure 4.1: Chapter 4 outline

4.1 INTRODUCTION

The goal of this chapter is to suggest and discuss the overall process model structure for an OSS migration and to provide a response to the third research question: *What are the key processes within an organisational open source migration?* This chapter is organised into the following sections: Section 4.2 explores the reasons why the CSIR migrated from proprietary sOftware to OSS, by analysing the detailed factors which have led to the adoption of the OSS migration project at the CSIR. Thereafter Section 4.3 summarises the steps which was followed to capture the processs knowledge of an OSS migration in the form of processes and subprocesses. Section 4.4 examines lessons learnt by those who were involved in the project, including the obstacles faced when migrating to an OSS environment. The chapter ends with concluding remarks. The rest of the chapter is outlined, as shown in Figure 4.1 above.

4.2 REASONS FOR MIGRATING FROM PROPRIETARY SOFTWARE TO OSS

The migration to OSS at the CSIR was motivated by several factors which were of importance to the organisation. The main reasons why the CSIR chose to migrate from a proprietary environment to an OSS environment was because OSS is: (a) free, it does not come with the expensive costs of buying software licences, (b) it comes with the freedom of allowing users to modify the source code to suit their computing needs, and (c) it is reliable and securable and it is less vulnerable to threats and viruses. The CSIR initiated the project due to those main reasons and established its own strategy for executing the migration. More on the CSIR migration plan is revealed in the next sections.

During this migration project the researcher observed the migration processes in the working environment, under the supervision of Vula team members. This enabled her to investigate, identify and capture all the processes that took place during the migration. More details on how the data was gathered to capture the process models for the OSS migration for this research study, are discussed in the next Section 4.3.

4.3 OSS MIGRATION PROCESS MODEL STRUCTURE

When this research study was conducted initially, the search done presented no existing procedures, tools or documentation for the OSS migration structure using process models. Therefore, this section briefly reports on the process followed to identify and capture the process model structure for the OSS migration. It was during this time that the data extracted from the previously described sources of data collection in Chapter 3 were used to reach the intended results of producing process models for the OSS migration project.

To choose the tools and the procedures that could be used to capture the process knowledge for the OSS migration, the first step was to do data gathering

at the CSIR to enable the identification and capturing of process models for the migration (see Chapter 3). The data gathering process began in 2007 and involved using the following resources to capture the OSS migration processes:

1. The web search of the literature, from the Vula project's website and other general websites about migrating from proprietary software to OSS. Example of the URL used for the Vula project is: https://vula.csir.co.za/index.php/Main_Page
2. The collection of the CSIR migration documentation which was already in place when data collection was conducted for this study.

Both the website and documentation of the Vula project consisted of planned stages of the migration, followed by the project team and the progress already made with regards to the migration. There is a possibility that during the course of the study there may have been some changes in data gathering dates due to changes in circumstances, however this has no impact on the data retrieved during the stage of data gathering. It is important to note that before the identification and capturing of OSS migration process models, several interviews were conducted with some of the Vula project team members.

The people who were interviewed in order to understand the framework of migration processes were the following Vula project team members, namely: Thomas Fogwill, Hennie Bezuidenhout, Randolph Verheij, Christa van der Merwe and Tlhogi Mokhema with whom the processes of the CSIR migration to OSS were discussed. The information received from these participants also helped to determine which CSIR units were in support of the migration and which ones were not. See the different scenarios of interview questions and responses captured in the two Tables below.

Interview template for Vula Team members

Unit:	Meraka
Interviewed:	Hennie Bezuidenhout
Date:	25 July 2008
Question 1:	Are there any processes that you feel were omitted from the structure?
Response:	Yes, five migration tracks were followed as part of the plan for this project. It is recommended that you visit the leaders of each track, talk to them to find out what each track entails. Then assess for yourself which processes were included or not.
Question 2:	Is this unit currently involved (or in support) of the CSIR OSS migration project?
Response:	Yes, in terms of the Technology and Research Track our unit is fully committed. Hennie suggested that I talk to Thomas regarding the Technology Track and talk to Randolph Verheij and Eric Basson with regards to the Implementation Track and Maintenance Track of the migration.
Question 3:	If YES to Question 2, when is the pilot or migration planned to take place in this unit?
Response:	The pilot for Meraka unit took place in 2008.

Unit:	Communication
Interviewed:	Christa van der Merwe and Tlhogi Mokhema
Date:	29 September 2008
Question 1:	Are there any processes that you feel were omitted from the structure?
Response:	Yes, have a look at the CSIR CEO/President e-mails and the one that was sent by Raynold Zondo, even Laurens Cloete to the whole of Meraka. Also read the Sciendaba internal newletters, Vula wiki website etc. to give you an idea of what was excluded.
Question 2:	Is this unit currently involved (or in support) of the CSIR OSS migration project?
Response:	Yes, in terms of communication, they still support the migration fully. The release of every Sciendaba always contains something about the migration and Tlhogi Mokhema is always invited to the Vula Leadership committee meetings every Thursday to represent his unit.
Question 3:	If YES to Question 2, when is the pilot or migration planned to take place in this unit?
Response:	The pilot for the Communication unit had been planned to take place after the CSIR Conference, said Christa. The conference was hosted from 17 - 18 November 2008. This was to shield the unit in case something goes wrong while the unit was busy preparing for the conference, they really did not want anything to interfere with the conference, so they prepared to go for training after the conference to get fully equipped for OSS and the migration.

These interviews were conducted with the aim of investigating further OSS migration processes, should there have been any left out.

The following steps, in Table 4.1, were therefore followed to identify and capture process models for the OSS migration project at CSIR.

Step	Step description	Tools / documentation used	Deliverable
1	Derive the high-level process model	Process listing with goals and resources	High-level process model
2	Refine the high-level process model to subprocesses	Subprocess and atomic process (sometimes referred to as subsubprocesses) listing	Subprocesses and atomic processes

Table 4.1: The procedure used for capturing OSS migration process models

As specified in Table 4.1, the high-level diagram is called a parent diagram and consists of main processes that can be easily broken down into child diagrams or lower-level diagrams, that is subprocesses and atomic processes. In this case the high-level diagram will be called a high-level process model diagram, while child diagrams will be referred to as lower-level process models diagram, to make it easier for everyone to understand the whole hierarchy of the OSS migration's processes and subprocesses. Subprocesses and atomic processes as shown in Table 4.1 represent a set of refined diagrams (lower-level process models), which were extracted from the high-level process models.

The first step in Table 4.1 indicates the extraction of high-level process models, by listing the key migration processes of the OSS migration project. From 2006 to 2008, which was the time this project was established as a research project, the aim was to capture the essence of migration processes that other organisations can learn from. It was during that time of the project that formal and informal interviews (pre-migration and post-migration interviews) were conducted with involved Vula project team members and CSIR employees. From the information gathered it emerged that it is important to have a generic migration process model structure that can assist those planning to embark on the same type of a project in future.

The second step refines the high-level process model diagram further into lower-level process models (that means high-level processes into subprocesses, subsubprocesses or atomic processes), which in this case include defining, identifying and capturing critical migration tracks which were followed as part of the migration plan during the project. As specified earlier in this same section, it was during this process that the Vula project website and documentation were utilised as resources to collect and confirm all migration processes that took place. This second step of refining high-level processes into subprocesses, consists of information gathered from those different migration tracks which were followed during the OSS migration project at the CSIR.

However in this section, only the essence of a high-level process model diagram and second-level of lower-level process models will be described, the rest of the refined atomic processes (that is from third level, fourth level to fifth level) can be viewed on the CD provided.

4.3.1 High-level diagram

In this section the focus will be on the identification of high-level process models of the OSS migration. The goal of a high-level process model diagram is to present first the main processes of the migration project. This high-level process model diagram is presented in Figure 4.2, it consists of the graphical representation of process names, inputs and outputs of summarised migration processes. First however, Table 4.2 consists of the diagrammatic description of each process's goals, inputs and outputs, to describe the intention of each process. The format followed in this section is providing the table first, followed by the process diagram and its associations' table of processes.

Process	**Input/output resources**	**Goal description**
Kick-start the project (A1)	Input: CEO Declaration Output: Project initiated	To prove the organisation's seriousness and commitment towards migrating to open source.
Form the project team (A2)	Input: Project initiated Output: Team formed	To make critical decisions during the implementation of an open source migration.
Announce the project publicly (A3)	Input: Team formed Output: Media coverage	To ensure that the public knows about this project.
Develop migration plan, divide the project into tracks (A4)	Input: Media coverage Output: Migration strategy	To help draw a roadmap for the current environment to transitional environment, thereby sub-dividing the project into tracks to allow much of the work to be done thoroughly by those given the responsibility for the task.
Communicate migration plans or project scope (A41)	Input: Migration strategy Output: Migration track	To create user awareness and excitement for changing to OSS.
Identify technology (A42)	Input: Migration strategy Output: Migration track	To check for alternative OSSs and the compatibility of such softwares against the current ones used by users.
Provide environmental training (A43)	Input: Migration strategy Output: Migration track	To provide users with the relevant training and build their skills to make them feel confident about the migration.
Prepare roll-out (A44)	Input: Migration strategy Output: Migration track	To prepare users and put them into action by installing some OSS-related applications on their desktops.

Plan and prepare maintenance (A45)	Input: Migration strategy Output: Migration track	To continue to provide all the help needed even after the completion of the migration.
Migrate scheduled users to an OSS desktop (A5)	Input: Migration strategy Output: Users migrated	To deliver an operational Linux desktop.
Support and maintenance (short term and long term) (A6)	Input: Users migrated Output: User support	To continue to provide all the help needed even after the completion of the migration.
Document lessons learnt (A7)	Input: User support Output: Migration completed	To provide guidance to other organisations planning to migrate to other distributions in future. How to go about the migration and to avoid any of the risks involved.

Table 4.2: High-level processes input, output resources and goals

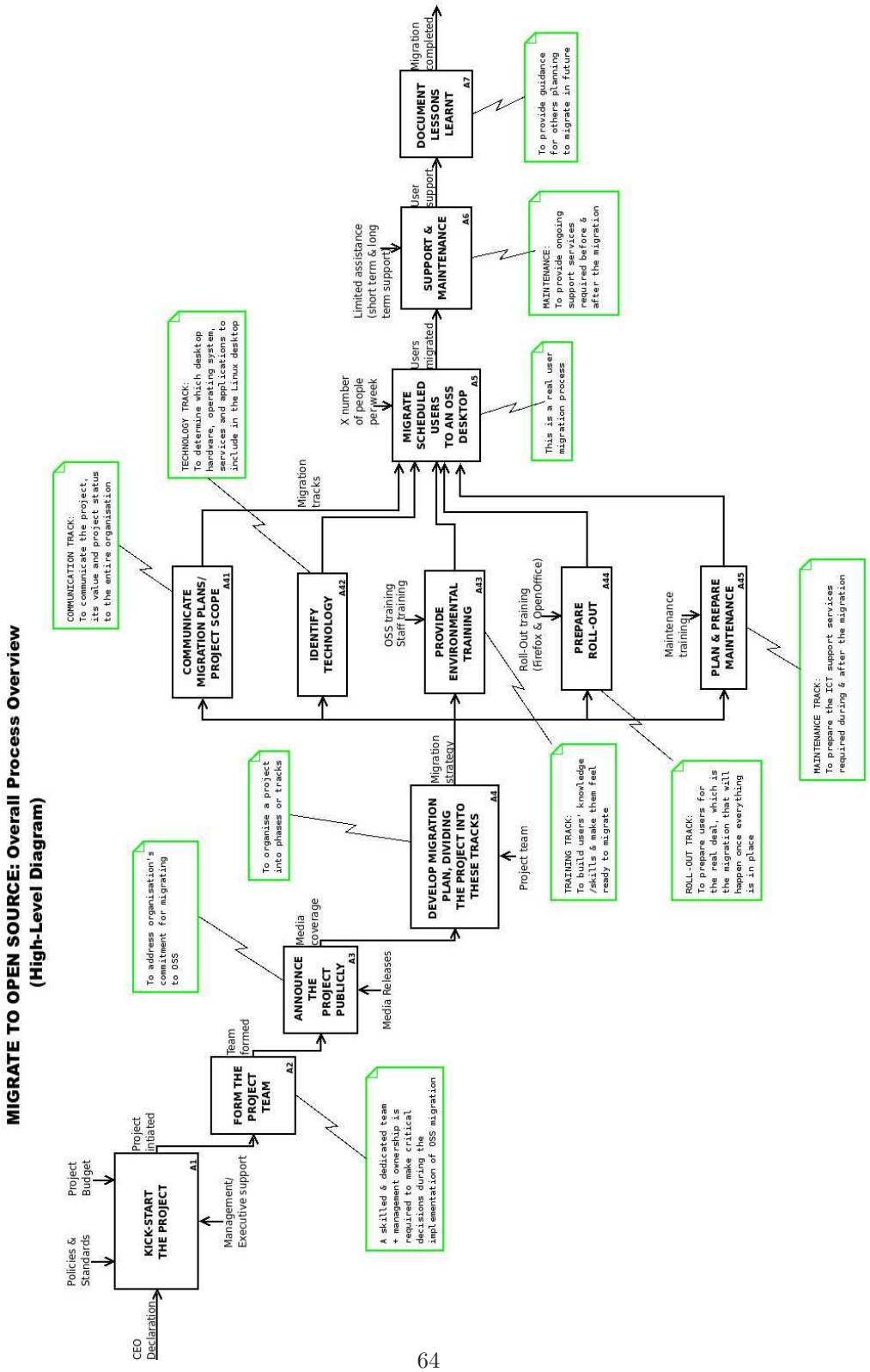

Figure 4.2: The OSS migration high-level diagram

As shown both in Table 4.2 and Figure 4.2, these are the identified and captured key migration processes used for deriving the high-level process model diagram. Each of these processes outlined in the high-level diagram were linked with their respective *input resources*, *output resources* and a *goal* - some with *mechanism resources* and *control resources* though the focus will be mainly on input and output requirements. Through the use of these input and output resources linking different processes with one another, the result was in the high-level process model diagram, the one shown in Figure 4.2. Briefly described below is the summary of the processes that took place concerning the migration as depicted in the high-level process model diagram.

It is shown in the high-level process model diagram that an output of one process can become an input of another process. For example, for the *Kick-start the project (A1)* process, 'CEO Declaration' was identified as an input resource while 'Project initiated' was the output resource. The output resource 'Project initiated' became an input resource for the next process which was *Form the project team (A2)*. It is possible to have more than one output in a process. It is also possible that other refined processes are without an input resource but this will be revealed as we continue to refine the high-level diagram into a lower-level diagram (that is second-level) in Section 4.3.2. In this case only one input and output were shown for each process. It is possible that there was other outputs for the *Kick-start the project (A1)* process, which may have been missed by the researcher, because they were not applied or documented. As depicted in Table 4.2 each process has its own goal, the *goal* of the *Kick-start the project (A1)* process was to provide assurance to the CSIR employees and its external stakeholders that the project had begun.

As for *Form the project team (A2)*, a reliable and committed team was required once the project was initiated. Among the team, a project leader was appointed to ensure that every member of the team participates and takes responsibility for the tasks assigned to them concerning the migration. The output resource 'Project initiated' of the previously mentioned process *Kick-start the project (A1)* was used as the input resource for the *Form the project team (A2)* process. Similarly the same procedure was followed for the *Announce the project publicly (A3)* process and all other processes that will be mentioned. This process's goal was to ensure that the public knew about this project and was kept up to date all the time with regards to its progress. 'Team formed' which was the output resource of the *Form the project team (A2)* process became the input resource of this process *Announce the project publicly (A3)*. Furthermore, 'Media coverage' which was the output resource of the *Announce the project publicly (A3)* process became the input resource of *Develop migration plan, divide the project into tracks (A4)* process. However, the *Develop migration plan, divide the project into tracks (A4)* process was refined further into child diagrams or second-level diagrams (that means subprocesses), which will be discussed in detail in Section 4.3.2. These subprocesses of the *Develop migration plan, divide the project into tracks (A4)* process included amongst others the migration tracks: Communication Track, Technology Track, Training Track, Roll-out Track and Maintenance Track.

The input resource 'Migration strategy' which was used for the five migration

tracks, produced the output resource 'Migration tracks'. This output resource acted as the input resource to the process *Migrate scheduled users to an OSS desktop (A5)* with the goal to allow the migration to be conducted successfully, without too many obstacles. Once desktops had been migrated successfully and users had enough training that prepared them for the real-deal in a working environment, support and maintenance continued to be offered to users on a daily basis. Migration knowledge was documented and the results will be publicly published at the completion of the project as clearly specified by the last process on the list *Document lessons learnt (A7)*, encouraging the sharing of all the lessons learnt with other organisations and businesses that might find it useful when considering migration to an OSS environment. The output resource 'Users migrated' of the *Migrate scheduled users to an OSS desktop (A5)* process acted as the input resource to the *Support and maintenance (long term) (A6)* process and 'User support' as its output which became an input to the last process *Document lessons learnt (A7)* producing the output 'Migration completed'.

Provided below in Figure 4.3 is the association table used to diagrammatically show the relationships between high-level diagram's processes, their input and output resources as depicted in a high-level process model.

Associations between input, output resources of a High level Process Model

		CEO Declaration	Project initiated	Team formed	Media coverage	Migration strategy	Users migrated	User support	Migration completed
A1	Kick-start the project	I	O						
A2	Form the project team		I	O					
A3	Announce the project publicly			I	O				
A4	Develop migration plan, divide the project into tracks				I	O			
A5	Migrate scheduled users to an OSS desktop					I	O		
A6	Support and maintenance						I	O	
A7	Document lessons learnt							I	O

Figure 4.3: Associations between resources and processes of a high-level process model

The results in Figure 4.3 represent how the processes in a high-level diagram are interconnected, whereby an output of one process can become an input of another without loosing the meaning of the context.

4.3.2 Second-level diagrams

In this section the goal is to present the refined lower-level process models (second level process models or subprocesses) of the above specified high-level process model diagram. Aim being to extract subprocess models linking to the high-level process model in order to complete the understanding of the OSS migration processes. The data used here were gathered from the interviews, questionnaires and literature reviews regarding the migration processes, as a guideline for the extraction of high-level process models discussed in the previous section and also in this instance for expanding the high-level process models further into second level process models.

4.3.2 (a) *Kick-start the project (A1)* process

From the data gathered during interviews, the *Kick-start the project (A1)* process was selected for inclusion in this case to illustrate that it is possible to derive this process further into subprocesses (see Table 4.3 and Figure 4.4). It is not only this process *Kick-start the project (A1)* that was refined into subprocesses, other processes also specified in Figure 4.2 were also expanded further into subprocesses as depicted in a high-level diagram. Therefore due to space limitations only the subprocesses (or second level diagrams of the high-level processes) will be shown, the rest of the levels (third, fourth and fifth) are available for viewing in the documentation provided in the CD.

Both Table 4.3 and Figure 4.4 presents the captured subprocesses of the *Kick-start the project (A1)* process. The link between the input, output resources and goal for each subprocess given in the *Kick-start the project (A1)* process are depicted in Table 4.3.

Process	**Input/output resources**	**Goal description**
Plan the migration (A11)	Mechanism: Project team Output: Project plan	To determine whether your organisation is ready to continue with the project of migrating to open source.
Create a brand for the project (A12)	Input: Project plan Output: Project brand	To promote and market the project.
Review work done by other organisations (A13)	Input: Project brand Output: Documentation review	To familiarise yourself with what other companies achieved that have done successful migrations.
Obtain organisational commitment to support the migration (A14)	Input: Documentation review Output: User support	To make users aware about the project.
Do awareness campaigns (A15)	Input: User support Output: Project awareness	To remind and motivate users about the seriousness of the project.

Demonstrate to users the planned OSS desktop (A16)	Input: Project awareness Output: Desktop prototype	To allow users to have a picture in their minds about the new desktop and its changes.

Table 4.3: Refined processes of the *Kick-start the project (A1)* process

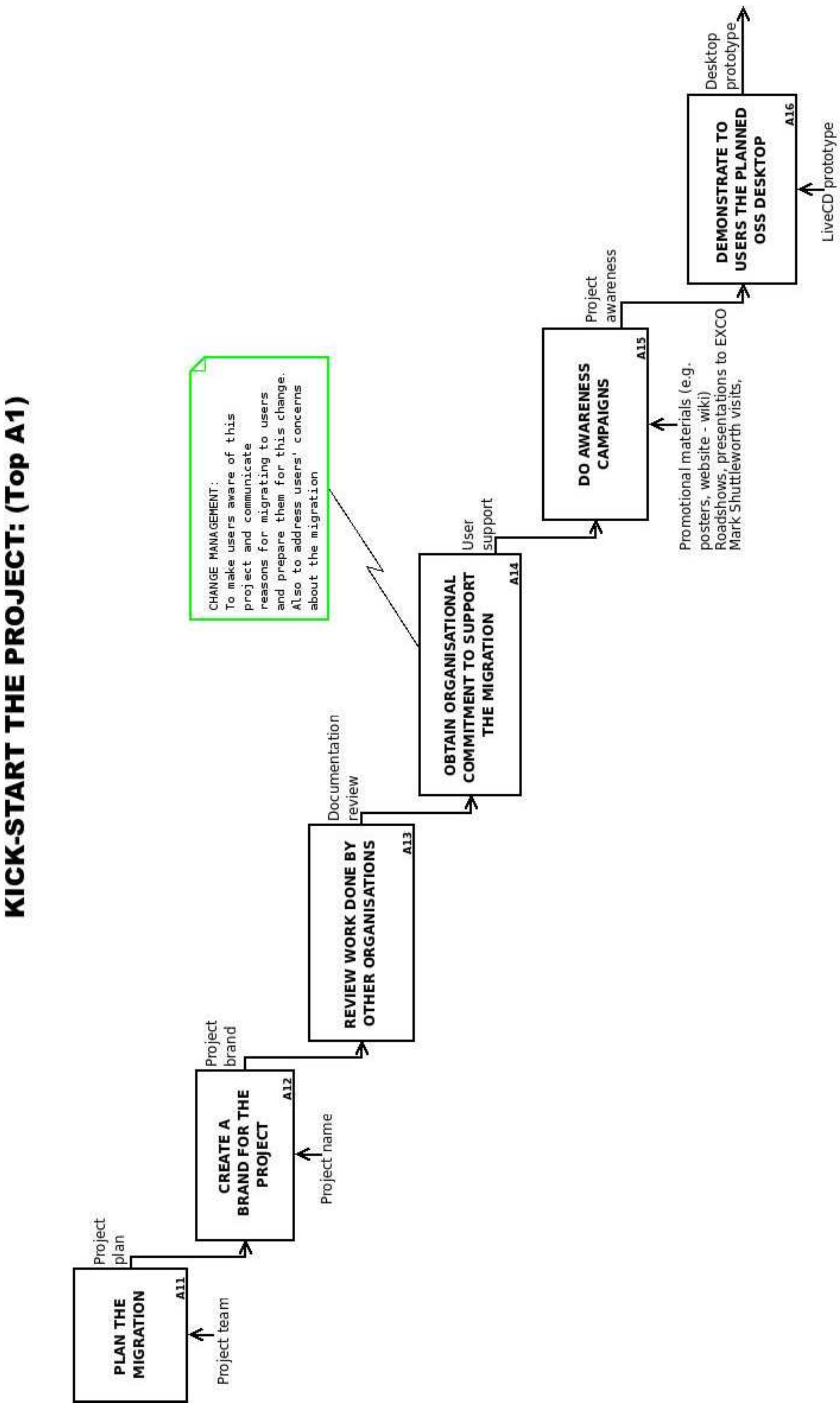

Figure 4.4: *Kick-start the project (A1)* subprocesses

To initiate the project Dr Sibisi who is the CSIR President/CEO together with the CSIR Executive made a decision in June 2006, to migrate the whole of the CSIR's desktops to an OSS environment. At this time the announcement about the CSIR migration to OSS was first heard by everyone within the organisation and outside. A plan was then drafted by the Vula project team members to ensure that everything runs smoothly, hence the subprocess *Plan the migration (A11)* identifying how the project was initially planned (or strategised). There was no input resource for this subprocess, only the output resource 'Project plan', which became an input for the *Create a brand for the project (A12)* subprocess. Establishing a brand name for the project was very important as it helped with promoting and marketing the project. This naming process was a task conducted by the Vula project team members. To make it more interesting they invited users (CSIR employees) to get involved by suggesting names for the project in the form of a competition. The user that came up with the best name won a prize and this is how the name 'Vula' came about.

The next subprocess followed was to *Review work done by other organisations (A13)* with 'Project brand' as its input resource. This scenario allowed the Vula project team to familiarise themselves with what other companies have done to have successful OSS migration projects. In the process the Vula project team discovered that there are some crucial procedures to follow when starting the migration project which included the subprocess *Obtaining organisational commitment to support the migration (A14)*. This subprocess involved motivating and getting users excited about the migration, while at the same time ensuring that all users will provide their continuous support towards the project. Furthermore, awareness campaigns were conducted often where people could see and read about it to advertise the project within the organisation, hence the subprocess *Do awareness campaigns (A15)*. This included using the Vula project website and e-mail as a point of communication to the organisation and printing posters or leaflets about the migration which were placed appropriately in all units of the CSIR to make users aware about the project. *Demonstrations to users about the planned OSS desktop (A16)* were also performed to allow users to see the prototype of the new OSS environment and to start experimenting with it.

Shown below are the relationships between the subprocess's input and output resources. The association table for the *Kick-start the project (A1)* process is given in Figure 4.5.

Associations between input, output resources for Kick-start the project (A1) process

		Project plan	Project brand	Documentation review	User support	Project awareness	Desktop prototype
A11	Plan the migration	O					
A12	Create a brand for the project	I	O				
A13	Review work done by other organisations		I	O			
A14	Obtain organisational commitment to support the migration			I	O		
A15	Do awareness campaigns				I	O	
A16	Demonstrate to users the planned OSS desktop					I	O

Figure 4.5: Associations between input, output resources for the *Kick-start the project (A1)*

4.3.2 (b) *Form the project team (A2)* process

The next scenario reveals decomposing the process *Form the project team (A2)* as depicted in a high-level process model diagram into subprocesses (or second level). Here a different process from the previous one is used as an example to illustrate the building of a process model on the second level, (see Table 4.4) highlighting the input, output and goal for each subprocess of the *Form the project team (A2)* process.

Process	**Input/output resources**	**Goal description**
Form the project team (A2)	Mechanism: Management, Executive support Output: Team formed	To help with the planning of the migration and ensure everything runs smoothly.
Identify those that will govern the project (A21)	Input: Team formed Output: Project governance	To select individuals that will manage or lead the project.
Divide project tasks within team members (A22)	Input: Project governance Output: Job description	To ensure that every member in a team is responsible for particular tasks given to him/her and completes it successfully.
Draw a schedule plan for the migration (A23)	Input: Job description Output: Migration plan	The migration plan will help to put things in order.

Table 4.4: Refined *Form the project team (A2)* process

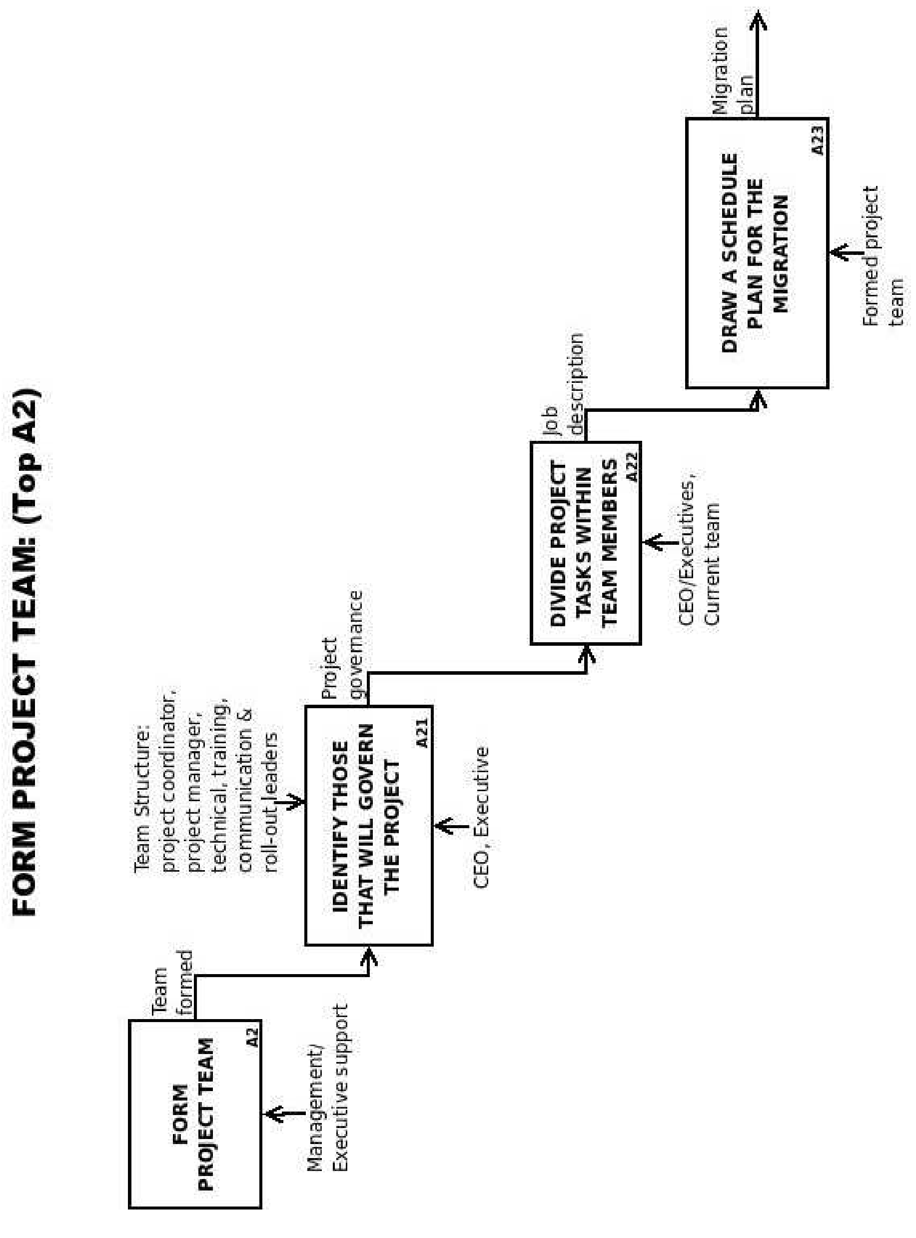

Figure 4.6: *Form the project team (A2)* subprocesses

The *Form the project team (A2)* process involved having a dedicated team, committed to planning the migration and ensuring that all the migration challenges and obstacles are taken care of as early as possible. This was the role played by the Vula project team during the CSIR migration to OSS. The team with the help and support of the CSIR President/CEO Dr Sibisi and Executives first appointed capable project leaders to manage the project. Responsibilities were shared amongst project members as the migration project was too large and could not be handled by one person alone. Each member was given a task based on their work experience or something they felt comfortable with.

Next, each of the subprocesses of the *Form the project team (A2)* process were linked with one another through their respective input resources, output resources and a goal. The results are reflected in Figure 4.7 in the form of a Table.

Associations between input, output resources for Form the project team (A2) process

		Team formed	Project governance	Job description	Migration plan
A2	Form the project team	O			
A21	Identify those that will govern the project	I	O		
A22	Divide project tasks within team members		I	O	
A23	Draw a schedule plan for the migration			I	O

Figure 4.7: Associations between input, output resources of the *Form the project team (A2)* process

4.3.2 (c) *Announce the project publicly (A3)* process

Furthermore, the process *Announce the project publicly (A3)* was also expanded into a second level but only resulted in two subprocesses *Announce the project internally and externally (A31)* and *Invite media (A32)*. The goal was to get more and more people informed about this OSS migration project, both internally and externally. Internally the information was communicated to the CSIR employees by the Vula project team, using e-mails and Vula project website as resources, while externally the media was invited to communicate the information to other interested parties outside the organisation that wanted to follow the project.

Presented below in Table 4.5 are the subprocesses inputs, output and goal resources of the *Announce the project publicly (A3)* process. This is followed by

the graphical diagram of its subprocesses in Figure 4.8 and the relationship among its input and output resources in Figure 4.9.

Process	**Input/output resources**	**Goal description**
Announce the project internally and externally (A31)	Mechanism: CEO/Project team Output: Public announcement	To make the open source migration project public.
Invite media (A32)	Input: Public announcement Output: Media Coverage	To get more and more people informed about this kind of technological migration.

Table 4.5: Refined subprocesses for the *Announce the project publicly (A3)* process

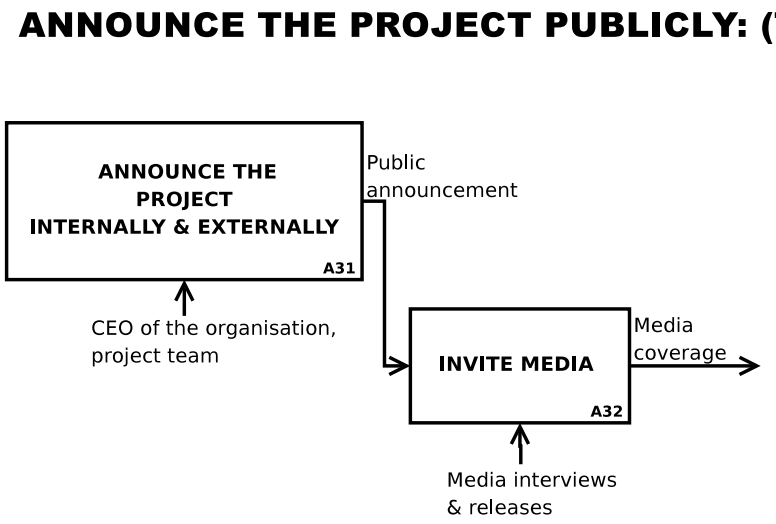

Figure 4.8: *Announce the project publicly (A3)* subprocesses

Associations between input, output resources for Announce the project publicly (A3) process

		Public announcement	Media Coverage
A31	Announce the project internally and externally	O	
A32	Invite media	I	O

Figure 4.9: Associations between input, output resources of the *Announce the project publicly (A3)* process

4.3.2 (d) *Develop migration plan, divide the project into tracks (A4)* process

Next the main focus is on the process *Develop migration plan, divide the project into tracks (A4)*, which is comprised of five migration tracks specified below. As the subprocesses of the *Develop migration plan, divide the project into tracks (A4)* process, these migration tracks were regarded as planned stages for the migration project, which were then followed when migrating from proprietary software to OSS. Without the migration tracks as part of the project, the migration route could have been much harder for the Vula project team to execute. The migration tracks are briefly listed and discussed below.

- Communication Track
- Technology Track
- Training Track
- Roll-out Track
- Maintenance Track.

As specified earlier, each of these migration tracks are the subprocesses of the main process *Develop migration plan, divide the project into tracks (A4)* shown previously in the high-level process model diagram in Figure 4.2. Each track was analysed on its own, with a number of sub-subprocesses (sometimes referred to as atomic processes) emerging from each. Due to space limitations only a summary of each migration track on a second-level is provided, the rest of the details about each of these tracks especially from the third level diagram to the fourth and fifth can be viewed in the documentation provided on the CD.

(i) Communication Track

For demonstration purposes we will first focus on the refinement of the Communication Track, see Table 4.6 highlighting the input, output and goal of each sub-subprocesses (or atomic processes) within the Communication Track subprocess. The Communication Track in summary is about the open communication that took place to make users aware about the migration project process (Figure 4.10).

Process	**Input/output resources**	**Goal description**
Develop a communication plan (A4)	Mechanism: Project team Output: Communication plan	To outline the communication approach that will be followed and to make the migration process and progress visible to everyone who will be affected.
Communicate migration plans or project scope (A41)	Input: Communication plan Output: Change communicated	To address the broader scope of change management, to ensure that your main stakeholders and users are well-informed about the migration.
Create user awareness (A411)	Input: Change communicated Output: User awareness plan	To promote and market the brand or project and also to persuade users and gain the organisational (users') support and commitment.
Communicate reasons for change (A4111)	Input: User awareness plan Output: Change motivated	To motivate users to accept this type of change.
Address users concerns (A4112)	Input: User awareness plan Output: User issues addressed	To understand users' worries regarding the migration.
Create positive momentum or motivation for change (A4113)	Input: User awareness plan Output: Excitement created	To make the migration process and progress visible to everyone who will be affected.
Build user knowledge or understanding of OSS (A4114)	Input: User awareness plan Output: Skills improvement	To familiarise users with the open source environment.
Prepare and distribute necessary information regarding migration progress to all (A412)	Input: Change motivated, User issues addressed, Excitement created, Skills improvement Output: Project update	To allow users to have a picture about the changes taking place.

Update users and top management internally with regular feedback (A4121)	Input: Project update Output: Internal feedback provided	To keep everyone in the organisation within the loop regarding the migration.
Communicate progress externally also (A4122)	Input: Project update Output: External feedback provided	To keep everyone especially external entities up to date about the project.

Table 4.6: Refined sub-subprocesses for the Communication Track subprocess

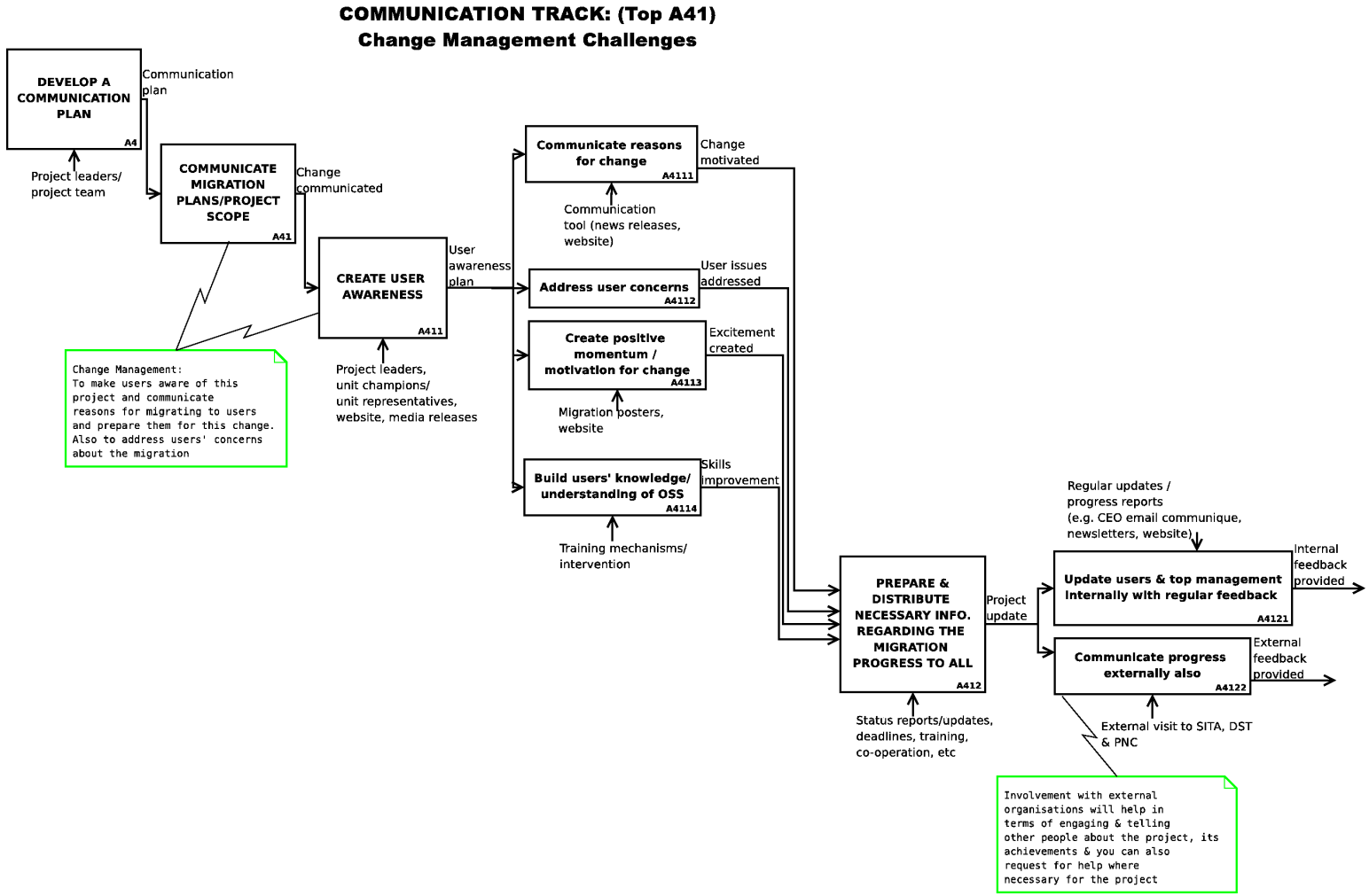

Figure 4.10: Communication Track sub-subprocesses

The Communication Track, represented in a high-level process model diagram by the subprocess *Communicate migration plans or project scope (A41)*, involves making users aware about the existence of the project. The goal of this track was to keep staff members of the CSIR up to date and encouraged about the migration, thereby communicating all the crucial information regarding the migration to all stakeholders. The Communication Track was broken down into several sub-subprocesses (or atomic processes), which included documenting the plan for this track and giving users an opportunity to address their concerns. The users' issues were submitted and then taken care of by the Vula project team for resolution. This track was led by a certain member of the Vula project team, who was appointed by the Vula steering committee to ensure that users concerns were handled and addressed in an orderly manner. The steering committee consisted of the CEO Dr Sibisi, the executives and Vula project managers. Presented below is the table of associations in Figure 4.11, depicting that a process can have more than one input, like in the case of this subprocess *Communicate and distribute necessary information regarding migration progress to all (A412)*.

Associations between input, output resources for Communication track (A41) subprocesses

		Input/Output Resources							
		User awareness plan	Change motivated	User issues addressed	Excitement created	Skills improv ement	Project update	Internal feedback provided	External feedback provided
A4111	Communicate reasons for change	I	O						
A4112	Address users concerns	I		O					
A4113	Create positive momentum or motivation for change	I			O				
A4114	Build user knowledge or understanding of OSS	I				O			
A412	Prepare and distribute necessary information regarding migration progress to all		I	I	I	I	O		
A4121	Update users and top management internally with regular feedback						I	O	
A4122	Communicate progress externally also							I	O

Figure 4.11: Associations between input, output resources of the Communication Track

(ii) Technology Track

This is the track which follows after the Communication Track, the Technology Track. In the case of the Technology Track which is represented by the subprocess *Identify technology (A42)* in a high-level diagram, the aim was to provide users with working open source applications that are comparable in every way to applications running on the proprietary desktop (Microsoft Windows). Therefore this track involved the design of the standard Vula desktop, whereby the selection and development of new technologies that are OSS compatible were considered. It is during this track that many technical aspects were tackled of planning the OSS desktop for the migration, such as completing an application inventory of the current Microsoft Windows desktops to help identify applications currently being utilised by users. A questionnaire was used in this instance to gather such information (see Section 3.3.2.4). This information was then used to help the Vula project team to easily identify the less-risk users who could be migrated first. Not only was this track about the design of the Vula desktop, it also involved the conversion of documents from the format of Microsoft Office to OpenOffice. This step began when Dr Sibisi the CSIR President/CEO announced the adoption of an ODF on 10 November 2006. The CSIR's adoption of ODF as a document standard was a first in South Africa.

Diagrammatic representation of the input and output resources of the Technology Track is given in Table 4.7. While, Figure 4.12 below graphically represents the atomic processes (or sub-subprocesses) of the Technology Track.

Process	**Input/output resources**	**Goal description**
Develop a plan to identify technological and business requirements (A421)	Mechanism: Project team Output: Desktop design plan	To determine which OSS desktop hardware, operating system, services and applications (relevant technologies) closely match the existing Microsoft Windows desktop.
Analyse the current architecture, infrastructure (A422)	Input: Desktop design plan Output: Infrastructure analysed	To address relevant technologies such as desktop hardware, operating system, services and applications that will be included in the Linux desktop.
Do an application inventory per user per machine (A4221)	Input: Infrastructure analysed Output: Application inventory completed	To help identify and classify the current desktop services and applications.
Categorise users according to their dependence on proprietary-based applications (A4222)	Input: Application inventory completed Output: Categorised users	To help the project team to easily identify the less-risk users who can be migrated first.

Investigate alternative OSS applications and assess compatability (A4223)	Input: Categorised users, Collected questionnaire results Output: Alternative applications identified	To determine if Linux does support those applications that are comparable to the ones currently used on the Microsoft Windows desktop.
Identify the graphical desktop, environment, distribution and applications that will be included in a standard desktop (A4224)	Input: Alternative applications identified Output: Desktop profile	To establish whether Linux can meet the needs of your users and to also check whether its worth being pursued in your organisation.
Design or develop pilot-base desktop (A423)	Input: Desktop profile Output: Desktop developed	To ensure that the designed Linux desktop is a close match to the existing Microsoft Windows desktop.
Test, customise and refine (or improve) the developed desktop (A4231)	Input: Desktop developed Output: Desktop customisation	To ensure that the designed Linux desktop profile meets users technological requirements by matching the existing Microsoft Windows desktop profile.
Prepare data centre servers, network infrastucture and support services (A4232)	Input: Desktop customisation Output: Network servers ready	To ensure that desktops connect to servers, storage, printers, and other network devices.
Establish a legacy data conversion centre and ICT support team (A424)	Input: Network servers ready Output: Facility established	To convert all the organisations' documents, spreadsheets and presentations to a new format (ODF).

Table 4.7: Refined sub-subprocesses of the Technology Track subprocess

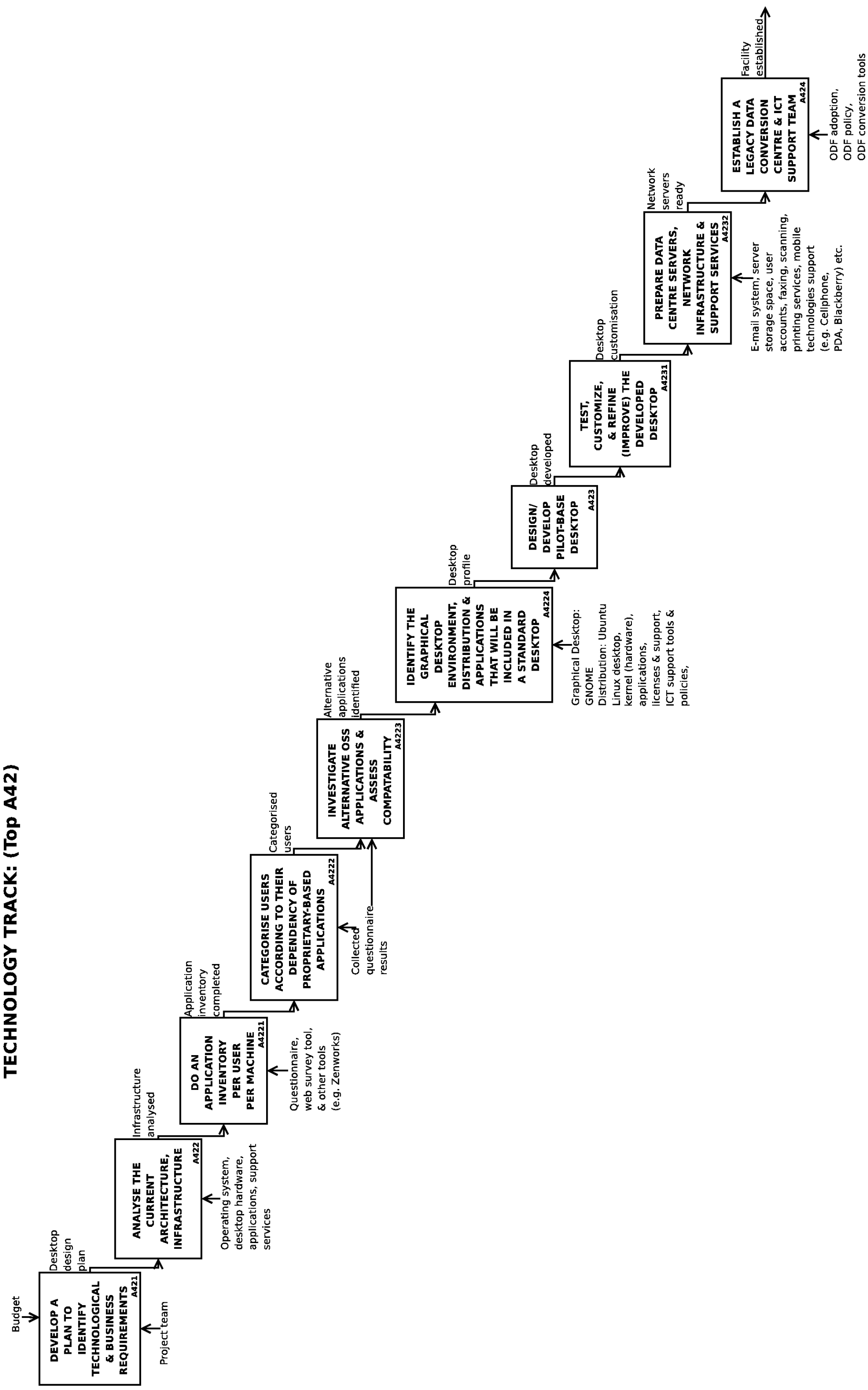

Figure 4.12: Technology Track sub-subprocesses

As shown in Table 4.7, the Technology Track comprises of the refinement of sub-subprocesses (or atomic processes) from third to fourth level. In this research study the processes were refined only until the fifth level and not beyond that. For instance the sub-subprocess *Test, customise and refine (or improve) the developed desktop (A4231)* which articulates how the designed desktop was improved, to ensure functionality of the whole desktop system was refined further into the fifth level which can be viewed in the documentation provided on the CD.

The associations table for the Technology Track sub-subprocesses was too large to be placed here, therefore it can be viewed together with the provided documentation on the CD.

(iii) Training Track

The Training Track represented by the subprocess *Provide environmental training (A43)* in a high-level process model diagram and involved selecting and training users who were deemed ready for migration and included the training of ICT technical staff (see Table 4.8 and Figure 4.13). Like all other tracks a plan was first implemented for this track, which included: having effecient resources in place such as a qualified training facilitator and a bigger computer lab that will accommodate as many users as possible during a training session. The training facilitator's responsibility was to teach scheduled users during a training session about Linux in general and to make it easier for them to adapt to the new Vula desktop environment. This included encouraging the scheduled individuals who underwent training not to resist change, but to allow it to happen and motivate their colleagues to support the organisation by partaking in the OSS migration.

Process	**Input/output resources**	**Goal description**
Provide environmental training (A43)	Control: Budget Output: Training plan developed	To prepare users by building their OSS knowledge.
Train users and technical staff (A431)	Input: Training plan developed Output: Trained user	To ensure that both users and technicians are familiar with the new environment.
Encourage self-training or allow users to experiment (A432)	Input: Trained user Output: Self-training tools	To allow users to train themselves about the new Linux desktop.
Provide practical or hands-on training (A433)	Input: Self-training tools Output: Practical training	To equip users with the necessary skills.
Call for early adopters (A434)	Input: Practical training Output: User pilots	To encourage volunteers that want to migrate.

Provide exclusive support to early adopters (A435)	Input: User pilots Output: Short-term support	To provide assistance immediately to migrated users for a limited period.

Table 4.8: Refined sub-subprocesses of the Training Track subprocess

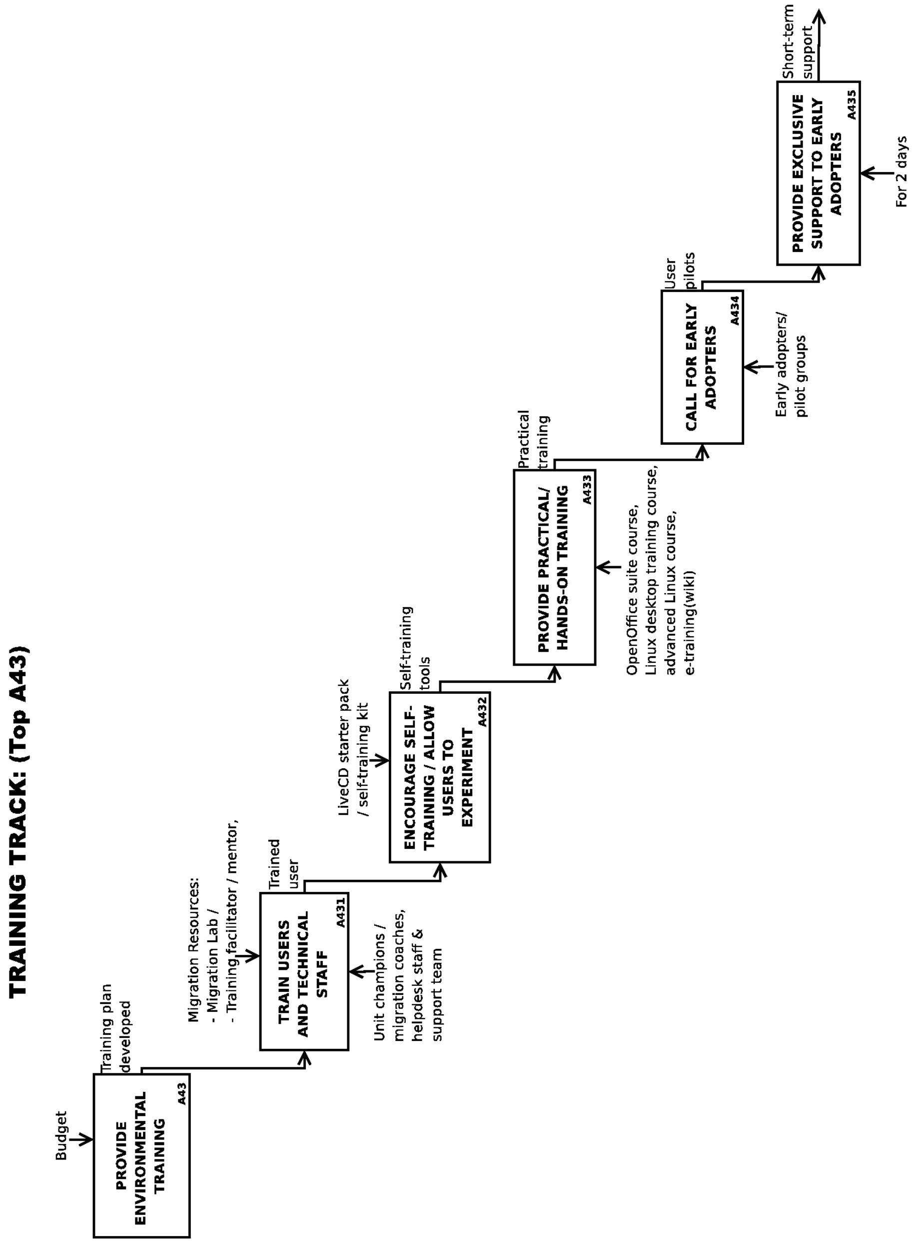

Figure 4.13: Training track sub-subprocess

When attending the two day training course, the users' machines were migrated to the Vula desktop in order to ensure the normal and productive use of the new desktop upon the return by the user after the training course. The training course's focus was to prepare users to proudly own the Vula desktop and navigate their way on that machine with comfort and ease. Presented below are the relationship between the Training Track's sub-subprocesses's (or atomic processes's) input and output resources.

Associations between input, output resources for Training track (A43) subprocess

		Training plan developed	Trained user	Self-training tools	Practical training	User pilots	Short-term support
A43	Provide environmental training	O					
A431	Train users and technical staff	I	O				
A432	Encourage self-training or allow users to experiment		I	O			
A433	Provide practical or hands-on training			I	O		
A434	Call for early adopters				I	O	
A435	Provide exclusive support to early adopters					I	O

Figure 4.14: Associations between input, output resources of the Training Track

(iv) Roll-out Track

The Roll-out Track represented by the subprocess *Prepare roll-out (A44)* in a high-level process model diagram, involved the preparation and handing over of an operating Vula desktop to the organisation. This official handing over was done when users were migrated to the OSS environment, in this case the Vula desktop. Like all other tracks, a plan describing the strategy for migrating the majority of users to an OSS desktop was addressed, at the same time ensuring that the intended expectations were met with regards to the CSIR migration to OSS. It was during this track (immediately after the adoption of ODF) that users' desktops within the organisation were equipped with relevant open source applications such as the new releases of OpenOffice and Mozilla Firefox. This was rolled out to the whole organisation to allow CSIR employees to familiarise themselves with some of these open source applications and to also undergo training to get used to the look and feel of the whole Vula desktop system. OpenOffice and Mozilla Firefox applications were replacements of Microsoft Office and Internet Explorer respectively and were used both on the Microsoft Windows desktop and OSS desktop (Vula). Hence the activity *Equip users with latest versions of OSS equivalence (A441).*

Without the readiness of the network infrastructure such as servers for e-mail or user accounts or any other support services, desktops will not be able to operate or connect to network servers, data storage, printers, and other network devices in either the OSS or Microsoft Windows environment. Therefore, it is essential to prepare network infrastructure and servers to enable desktops (both Vula and Microsoft Windows) to interoperate with other desktop operating systems as well as with directory and authentication services. At the same time desktops must be managed and supported by qualified personnel (ICT technical staff). Hence the generic activity *Ensure servers, network infrastructure and support services are ready for migration (A442).*

During this track users who were deemed ready for migration or those who were easily identified by the project team as less-risk users were migrated first, hence the sub-subprocess *Conduct pilot-courses (A443).* This small number of users sometimes referred to as early adopters were prepared for pilots one at a time. Conducting pilots was part of testing the Vula desktop and its associated applications, but with a small number of users. These pilots allowed users to give feedback regarding the new look and feel of the desktop, which in turn helped the team to rectify all kinds of mistakes. Articulating the procedure used for scheduling users to undergo training, a refined sub-subprocesses *Select users deemed ready to migrate (A4431)* and *Schedule selected users for migration (A4432)* provide a description on how these users who were deemed ready to migrate were initially scheduled for training. Furthermore the sub-subprocesses also states the aim of the Vula project team which was to achieve the maximum number of migrations of 40 desktops during a typical week. At first, this goal was doable. A lot of users volunteered to participate in the migration due to the excitement they were experiencing about this project. However, as time passed it was a challenge for the project team to keep to this number of users and their support from decreasing.

Clearly articulating the strategy on how desktops were prepared for the migration is the process *Migrate scheduled users to an OSS desktop (A5)*, described in page 98.

Process	**Input/output resources**	**Goal description**
Prepare roll-out (A44)	Mechanism: Project team Output: Developed plan	To describe the strategy for migrating the majority of users to a Linux desktop.
Equip users with latest versions of OSS equivalence (A441)	Input: Developed plan Output: Software rolled-out	To ensure that they get used to the look and feel of OpenOffice and Mozilla Firefox as a replacement for Microsoft Office and Internet Explorer.
Ensure servers, network infrastructure and support services are ready for migration (A442)	Input: Software rolled-out Output: Network servers prepared	To ensure that desktops connect to servers, storage, printers and other network devices.
Conduct pilot-courses (A443)	Input: Network servers prepared Output: Training strategy implemented	To prepare users for change and to allow them to freely give feedback regarding the new desktop that will help the team with rectifying the identified mistakes.
Select users deemed ready to migrate (A4431)	Input: Training strategy implemented Output: User readiness	To encourage more users to come forth and migrate and at the same time increasing the number of already migrated users.
Schedule selected users for migration (A4432)	Input: Training strategy implemented Output: Scheduled users	To ensure that the selected individuals or users are migrated.
Communicate process to selected users regarding the migration (A4433)	Input: Training strategy implemented Output: Project info. communicated	To allow users and even training technicians to know of any necessary information they will need in order to complete the migration.
Assign trained technical ICT staff with the responsibility to do the actual migration or installation (A444)	Input: User readiness, Scheduled users, Project info. communicated Output: Assigned personnel	To ensure that no data is lost during the migration.

Migrate scheduled users to an OSS desktop (A445)	Input: Assigned personnel Output: Users migrated	To ensure that everything work efficiently as expected.
Continue with the conversion of templates and standard documents (A446)	Input: Users migrated Output: Documents converted	To allow users time to get used to a new format.

Table 4.9: Refined sub-subprocesses of the Roll-out Track subprocess

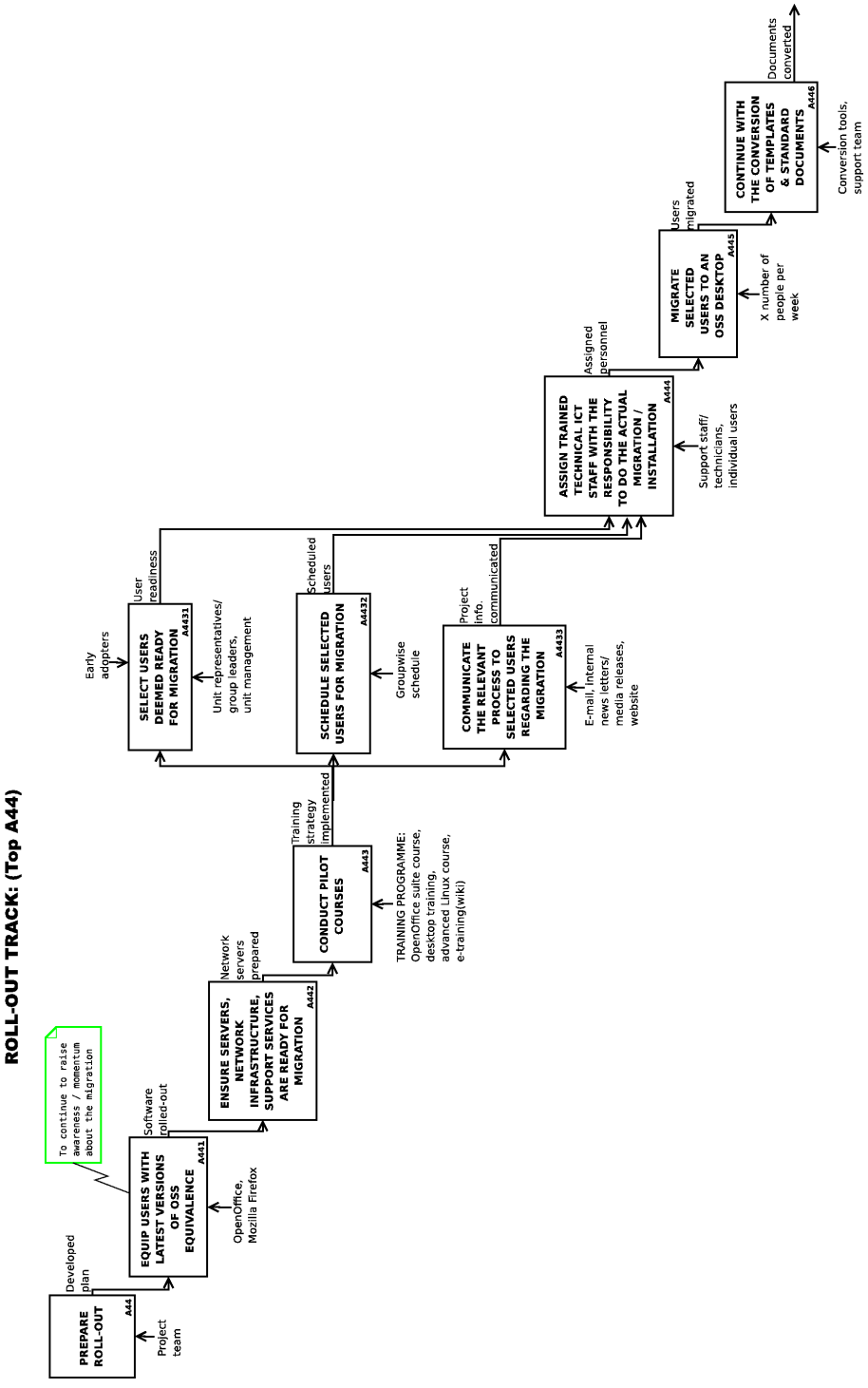

Figure 4.15: Roll-out Track sub-subprocesses

Due to limited space, the associations table for the Roll-out Track's sub-subprocesses (or atomic processes) could not fit. However it can be viewed in the document provided on the CD.

(v) Maintenance Track

The Maintenance Track, represented by the subprocess *Plan and prepare maintenance (A45)* in a high-level process model diagram, involves preparing how support will be provided to users once they have been migrated (or once the above steps have been completed) and the user is at ease with operating the Vula desktop. A plan was put into place for how users who are now utilising a Vula desktop will be supported and also how the support for the current Microsoft Windows desktop users will be continued. The goal was to ensure that everything was ready and continued even after the migration. Therefore it is crucial that ICT technical staff (or technicians) be trained in advance to perform this specific job. Table 4.10 identifies the sub-subprocesses (or atomic processes) inputs and outputs associated with the Maintenance Track.

Process	**Input/output resource**	**Goal description**
Plan and prepare how maintenance will be done (A451)	Output: Plan developed	To determine how the users who are now utilising different desktop environments will be supported.

Table 4.10: Refined sub-subprocess of a Maintenance Track subprocess

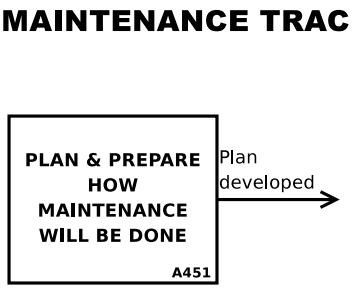

Figure 4.16: Maintenance Track sub-subprocess

Since only one activity is shown in the Maintenance Track, the association table revealing relationships among the input, output resources of the Maintenance Track was not created.

4.3.2 (e) *Migrating scheduled users to an OSS desktop (A5)* process

For the *Migrate scheduled users to an OSS desktop (A5)* process, various steps were followed to illustrate the process of the actual migration done on users' desktops, performed by qualified ICT technical staff while users were on training. It is important to note that the project leader initially communicated all the necessary information to users before they were migrated or underwent training. The aim was to help eliminate any confusion and surprises which could potentially affect the success of the migration for scheduled users. These users were scheduled by making appointments with the individuals through a Groupwise e-mail to inform them of the dates, times and area where the training session was taking place. These details were provided in advance to users, to allow users to choose the migration slots (or dates) that would suit their availability.

Process	**Input/output resources**	**Goal description**
Communicate process to scheduled users (A51)	Input: Scheduled users with names, dates per week Output: Process communicated	To eliminate any confusion and surprises which can potentially affect the success of the project.
Confirm user training completed (A52)	Input: Process communicated Output: User trained	To ensure that a user undergo training and that users can function productively in the OSS environment on completion of the desktop migration.
Gain exclusive access to collect desktop (A53)	Input: User trained Output: PC fetched	To prepare desktops for installation.
Migrate desktop (A54)	Input: PC fetched Output: X number of migrated PCs	To successfully complete the migration.
Record progress (A55)	Input: X number of migrated PCs Output: Total number of migrated users	To keep track of how many users have migrated and how many haven't migrated.
Release desktop (A56)	Input: Total number of migrated users Output: PC delivered	To enable the user to proceed with work and apply skills learnt from the user training with possible assistance from the coaches.
Provide limited assistance and support (A57)	Input: PC delivered Output: Up and running user	To assist users with all aspects of an OSS desktop, should there be any problems experienced by users.

Hand over to ICT operations (A58)	Input: Up and running user Output: Support services provided	To provide standard support and address all issues that could have arose after the migration by working hand-in-hand with coaches.

Table 4.11: Refined subprocesses of *Migrate scheduled users to an OSS desktop (A5)* process

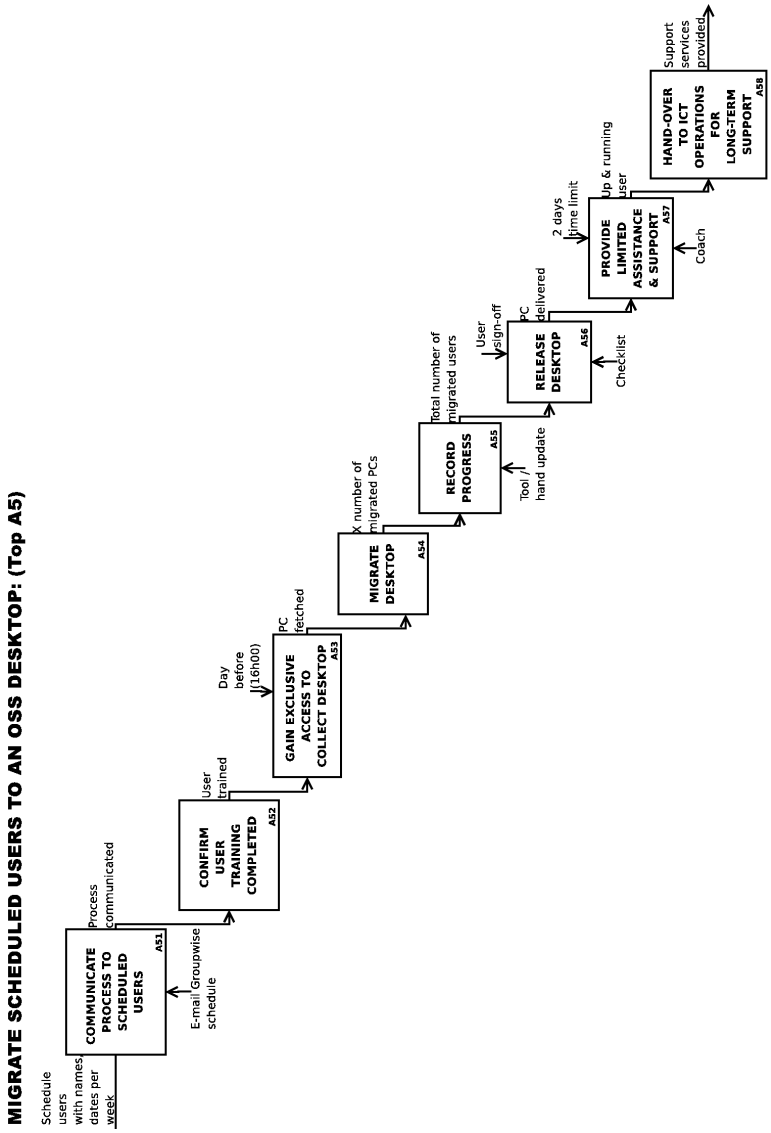

Figure 4.17: *Migrate scheduled users to an OSS desktop (A5)* subprocesses

This association table (Figure 4.18) is used to show the input and output relationship of the *Migrate scheduled users to an OSS desktop (A5)* process on the second level.

Associations between input, output resources for Migrate scheduled users to an OSS desktop (A5) process

		Scheduled users with names, dates per week	Process communicated	User trained	PC fetched	X number of migrated PCs	Total number of migrated users	PC delivered	Up and running user	Support services provided
A51	Communicate process to scheduled users	I	O							
A52	Confirm user training completed		I	O						
A53	Gain exclusive access to collect desktop			I	O					
A54	Migrate desktop				I	O				
A55	Record progress				I	I	O			
A56	Release desktop						I	O		
A57	Provide limited assistance and support							I	O	
A58	Hand over to ICT operations								I	O

Figure 4.18: Associations between input, output resources of the *Migrate scheduled users to an OSS desktop (A5)*

4.3.2 (f) *Support and Maintenance (A6)* process

After the completion of the migration and training of scheduled users, more assistance was provided to users by deployed migration coaches who were fully skilled with the Linux desktop background. They were placed for a period of two days in respective units to assist struggling users immediately. Thereafter, once the users were comfortable with using the Vula desktop, the responsibility was handed over to the ICT support services to continue with standard support and maintenance on a long-term basis and address all issues that needed to be taken care of soon after the migration (see Table 4.12 and Figure 4.19).

Process	**Input/output resources**	**Goal description**
Provide ongoing support services even after migration (A61)	Mechanism: Helpdesk, desktop support staff (1st line and 2nd line support) Output: Support services	To ensure that everything runs smoothly even after the migration.

Make system enhancements, upgrade OSS applications with latest updates (A62)	Input: Support services Output: System updates	To ensure that the whole system is updated with latest software versions.
Ensure there is still positive and continuous commitment to change (A63)	Input: System updates Output: Continuous project momentum	To ensure that project keeps going on and the rest of the organisation is migrated.

Table 4.12: Refined sub-subprocesses of the *Support and Maintenance (A6)* subprocess

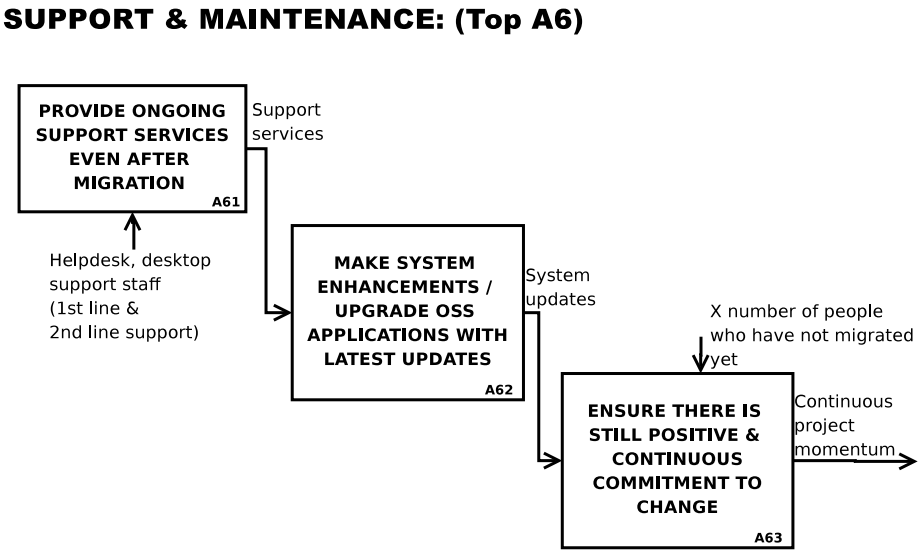

Figure 4.19: *Support and Maintenance (A6)* sub-subprocess

Next is the association table listing all the sub-subprocesses (or atomic processes) for the subprocess Support and Maintenance (A6).

Associations between input, output resources for Support and Maintenance (A6) process

		Support services	System updates	Continuous project momentum
A61	Provide ongoing support services even after migration	O		
A62	Make system enhancements, upgrade OSS applications with latest updates	I	O	
A63	Ensure there is still positive and continuous commitment to change		I	O

Figure 4.20: Associations between input, output resources of the *Support and Maintenance (A6)* subprocess

4.3.2 (g) *Document lessons learnt (A7)* process

Once all the steps regarding the migration were completed, the Vula project team thought it was useful to document the lessons they learnt during the migration. They thought this documentation could act as a guide to those organisations that may consider migrating to an OSS environment in future, be it educational institutions or businesses. It was also seen as a good opportunity for the CSIR to expand relationships and collaborations with different types of organisations planning to migrate to an OSS environment. The document will be published at the completion of the CSIR migration and will act as a guide on how to perform different processes regarding the OSS migration project.

Process	**Input/output resources**	**Goal description**
Document the migration and lessons learnt (A71)	Output: Migration documented	To guide others on how to go about the migration and to avoid any risks involved.
Review and share lessons learnt with other organisations planning to migrate to OSS in future (A72)	Input: Migration documented Output: Migration lessons circulated	To help other organisations migrating to a Linux desktop. Even if they are adopting a different distribution they could make use of the document.

Table 4.13: Refined subprocesses for the *Document lessons learnt (A7)* process

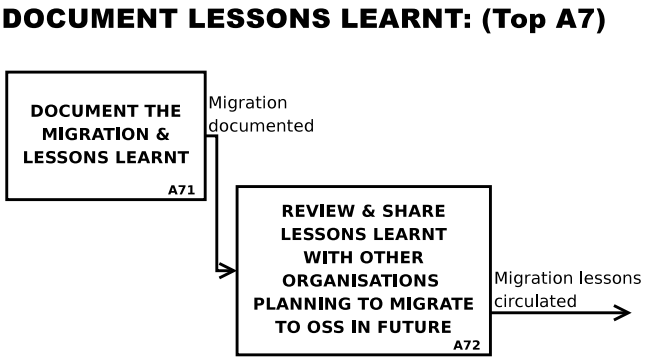

Figure 4.21: *Document lessons learnt (A7)* subprocesses

Listed below in a table (Figure 4.22) are the associated inputs and outputs of the *Document lessons learnt (A7)* process.

Associations between input, output resources for Docum lessons learnt (A7) process

		Migration documented	Migration lessons circulated
A71	Document the migration and lessons learnt	O	
A72	Review and share lessons learnt with other organizations planning to migrate to OSS in future	I	O

Figure 4.22: Associations between input, output resources of *Document lessons learnt (A7)*

4.4 FINDINGS

Like with any other type of migration, several problems were encountered when constructing the high-level process model for the CSIR migration to OSS. The findings in this section reveal the experiences encountered by the researcher during the identification and capturing of OSS migration process models.

4.4.1 Lessons learnt from identification of process models

The lessons learned by the reseacher while participating in the OSS migration project and deriving process models will be discussed. A high-level process model was done when it was realised that there was a need for process models that can suggest not only the processes of the OSS migration, but that will also help to indentify and capture generic migration processes that will assist with the planning and implementation of an OSS migration project. A decision was made to limit the scope of the study to process modelling, which resulted in the research question focussing only on the process model structure of an organisational OSS migration. Therefore, the lessons learnt that are described later in this section revolve around the following issues: interviews and interaction between the researcher and participants (that is Vula project team members and CSIR staff) during the identification and capturing of OSS migration process models.

It took a lot more time and resources than expected to extract the process models for the OSS migration project, because there were not enough examples showing the process model structure for the migration. That was not the only problem experienced, Figure 4.23 graphically shows two more obstacles encountered during the extraction of process models. These obstacles are described below, to help the reader understand what took place when deriving process models for an OSS migration project.

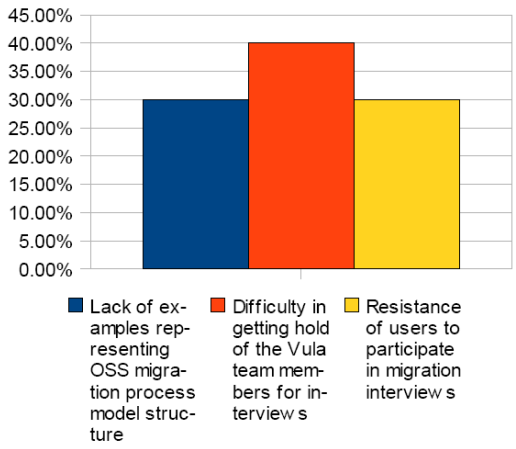

Figure 4.23: Problems encountered during the extraction of OSS migration process models

As shown in Figure 4.23, a number of problems were identified by the researcher by being involved in the Vula migration project as an observer when capturing the migration processes. First it was difficult to get hold of Vula project team members for interviews and they were the ones that could provide an explanation about the planned stages of the migration and describe how they intend to execute the migration. The goal for conducting these interviews was to establish whether or not the interviewed project team member agreed with what had been documented already about the migration plans. However, every member of the team were swamped with a lot of work at that stage so it was not easy to get an appoinment scheduled with them. These members all wanted to ensure that the actual migration takes place as anticipated on time and successfully. First an e-mail was sent to schedule appointments with some of the project team members. See a brief summary of the interview questions and responses captured in Tables in page 62 and 63 with some of the Vula project team members who finally made time to talk about their knowledge regarding the project.

It was also difficult at first to get the appointments scheduled with certain project team members because some were very unsure about my work and reasons for interviewing them. After I confronted them and assured them that there would be no threating questions directed at them, I was allowed to interview them. The interviews begun with a friendly discussion in which the respondents were allowed to say anything that was on their minds that could give a clear description about the migration. This approach was used to keep the respondents at ease, to show that I was interested in listening to the person I was talking to and wanted to get as much information about the migration as possible.

While deriving process models, a tool for modelling the processes graphically

was put into consideration. At first a feasibility study was conducted on different tools that were OSS-related that could be used for modelling the processes. The tool chosen was DIA and was used to graphically represent the process models and its subprocess models using a standard process modelling notation called IDEF0. This application, DIA, already exists on the Vula desktop.

The theory gathered from the interviews about the migration activities was used as a guideline for deriving process models. The steps specified in Table 4.1 in Section 4.3, which were defined in 2008 for the identification and capturing of process model structure for the OSS migration were also utilised during the process. The aim was to ensure that the high-level process model is produced together with its refined subprocess models. At first following and achieving those steps in Table 4.1 seemed unrealistic. However, with the assistance of the tool chosen to graphichally represent the processes, the steps were easier to follow though it took longer than expected to reach the intended results.

In contrast to the experiences with Vula project team members, scheduling appointments with migration early adopters for interviews were not much of a struggle. After being migrated, a 20 minute appointment was set up for few days after. These appointments were with users who were prepared to talk about their individual experiences with regards to the migration. Those experiences were then documented and can be viewed in the documentation provided on the CD.

4.4.2 Time

The time spent on data collection was longer than the time spent on modelling the processes. The process for gathering data was a tedious one because the longest period was spent on understanding the activities of the CSIR OSS migration project. This was done by first conducting interviews with different Vula project team members and it took a lot of time to meet with them and it was difficult to have appointments scheduled with them due to their workload.

Also, it took a bit of time to get familiar with the different CSIR units which were in support of the migration project and to realise who were resisting. To know this was a basic requirement as I had to set up meetings with the unit champions. The name 'unit champions' was used to refer to individuals who were selected in their different units by the Vula project team to assist with the units' users' issues concerning the migration.They had to ensure that everyone within their units were familiar with the migration project and up to date with its progress. In a short period of time the initial idea of having unit champions in place was scrapped from the migration plan due to several reasons. The name 'unit champions' no longer existed and was replaced by the word 'migration coaches' and their tasks were more or less the same; to promote the migration process and provide limited assistance to users.

Though the process took longer than expected, once the data was captured it took about a year or less to model the involved process models and refine them into subprocesses and sub-subprocesses (or atomic processes) (that is sec-

ond level, third level until the fifth level), and thereafter to extract the generic process models from the captured process models, which will be the next step discussed in Chapter 5.

At least there were no financial implications with the time spent to model processes of the OSS migration project.

4.5 SUMMARY

In this chapter the process models of the CSIR migration to OSS were presented, with the deliverable being the high-level process model diagram, which was derived further into lower level diagrams. In section 4.4, the interviews which were conducted with several Vula project team members proved to be valuable as they provided insight on the whole process which was followed specifically for the CSIR migration to OSS.

In Chapter 5, the extraction of generic process models (that is process reference models) will be discussed with regards to the contribution of the evidence that is found in this chapter, together with the validation or analysis of results suggested by the field experts of the Vula project team.

Chapter 5

CONTRIBUTION: THE GENERIC OPEN SOURCE PROCESS REFERENCE MODELS

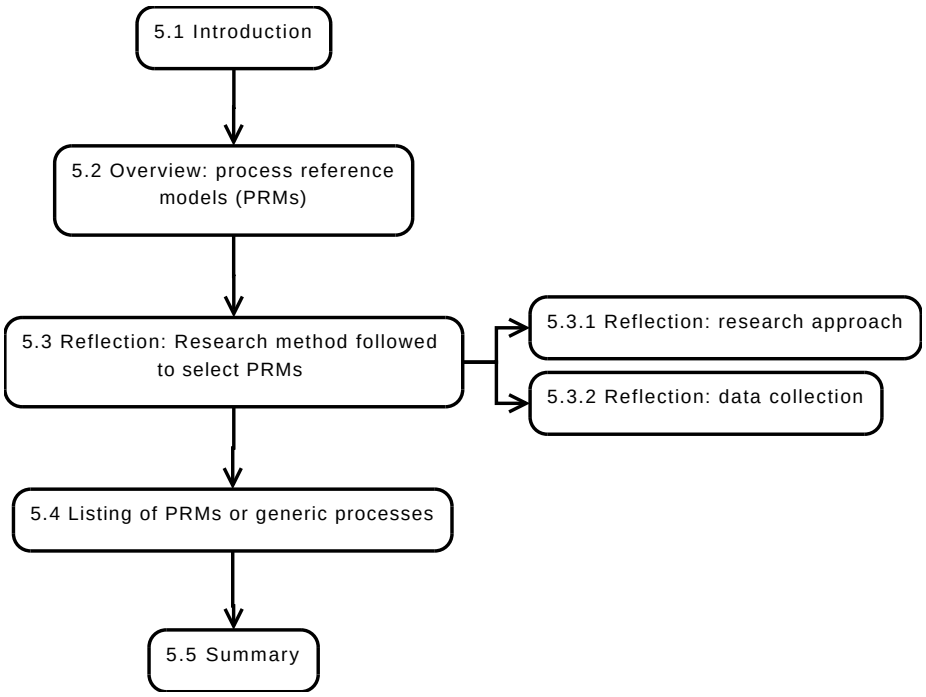

Figure 5.1: Chapter 5 outline

5.1 INTRODUCTION

The previous chapter addressed the third research question for this study: *What are the key processes within an organisational open source migration (OOSM)?* Data gathering was conducted at the CSIR to derive the OSS migration process models and were modelled using a standard process modelling notation called IDEF0. This resulted in the deliverable process models for an OSS migration in Chapter 4 which were further refined into a set of subprocess models.

In this chapter the focus is on the suggestion of process reference models extracted from a set of process models articulated in Chapter 4. For this chapter, the research question *What are the process reference models that are essential for an OOSM?* is addressed. The purpose is to list the process reference models identified during the CSIR migration project to OSS. To address this research question, the generic process model structure for the OSS migration will be shown and described in this chapter. Section 5.2 of this chapter provides a summary of what a process reference model is, followed by Section 5.3 reflecting on the research approach, research method, data collection and analysis techniques used for this study. Section 5.4 lists the generic process models. The chapter concludes with a summary in Section 5.5.

5.2 OVERVIEW: PROCESS REFERENCE MODELS

As specified earlier in Section 2.5, a process reference model is a term used to refer to a generic process model structure; a library of individualised process models in a certain application domain showing the graphical flow of processes and subprocesses as well as their relationship. These individualised process models can be reused by different organisations and companies to perform their business functions in a similar way without having to design their own from scratch (Rosa et al., 2005).

Without the use of process reference models for this study, which are often used to capture the common activities, roles and resources of any process in a certain environment and thereafter adapt them in another (Rosa et al., 2005), it would have been difficult to provide a generic library of process models for an organisational OSS migration project. Thus the process knowledge gained from the study enabled the extraction of process reference models for a typical OSS migration project. These process reference models are outlining all the repeatable organisational OSS migration processes (Pawlak et al., 2004), which can assist other organisations and companies migrating to OSS to do so successfully. Project managers and their teams can then better execute and plan these types of projects.

5.3 REFLECTION: RESEARCH APPROACH FOLLOWED TO IDENTIFY PROCESS REFERENCE MODELS

This section briefly addresses the method chosen to collect and identify previously listed process models of the CSIR migration to OSS in Chapter 4.

5.3.1 Reflection: research approach

As previously stated in Chapter 1 and 3, a case study approach was used for the data collection of this study, together with a systematic research approach; Five phases can be followed during the identification of a process model structure and only three of those phases were utilised for this research. The process models were modelled using a standard process modelling notation called IDEF0.

In essence, the TOP-DOWN procedure was used to list all the migration processes which took place during the project from high-level process models to lower-level process models. The TOP-DOWN approach means that the focus was on identifying the high-level diagram first for the OSS migration processes, followed by a thorough assessment and refinement of high-level processes into child diagrams or subprocesses. Thereafter, selecting processes which seemed generic was the focus of this chapter. In theory the process models which will be identified in this chapter can be reused, but in practice these generic process models will have to be tested first against any another environment to see how effective they are. However, this will be considered as future work arising from this study. For the present scope of this study, all generic processes were labelled in the form of *red text*. Those that are partially generic and non-generic were labelled in *blue text*, to be easily identified when discussed in detail in Section 5.4.

5.3.2 Reflection: data collection

The methods of the case study approach which were fit for investigating the CSIR migration activities to OSS included observing the migration right from the working environment, interviewing project managers, project team members and users who were involved in the migration and reviewing in detail all the documentation that had to do with the migration, such as the questionnaire and other examples of OSS migration projects already conducted worldwide.

As stated in Section 3.3.2.1 and shown in Tables in page 62 and 63 pre-migration interviews were conducted with several senior members of the Vula project team to help the researcher understand the process involved in migrating users to an OSS environment. The first interview was with the project manager Hennie Bezuidenhout, the second with technical leader Thomas Fogwill and the last interview was with the Communications' unit managers Christa van der Merwe and Tlhogi Mokhema. Thereafter post-migration interviews were held with numerous migrated users to note their experiences. All the questions and responses

of the post-migration interviews are available on the included CD. The interview guide comprised of the following sections: purpose of the interview, names of the interviewee, interviewer guideline questions and the table for interview responses. In the documentation provided on the CD, the interviewees elaborated on their experiences, how they viewed the migration project and whether it benefited them or not.

All the interviews were conducted in person, one on one and were recorded. Thereafter the responses were documented and the report was given to the ICT manager to work through, together with the Vula team to implement changes where necessary. The format and process of the interview was at all times kept on track by the interviewer.

The questionnaire also contributed to the collection of data for this research, based on participants' responses which were then captured and analysed by the Vula project team. The observations and literature review were also valuable tools used for gaining knowledge about the activities of the migration to OSS.

After using the CSIR as a case study, it is strongly believed that the generic process models (process reference models) can be a useful guide to other organisations venturing into the same type of a project. However, this is something that must be verified by another research study that can be done in future, as it was not part of the scope for this dissertation. For future research it will be useful to verify the findings of the CSIR migration to OSS in a similar project but in a different organisation.

5.4 LISTING OF PROCESS REFERENCE MODELS

This section provides a list of extracted process reference models which were captured by the researcher from the CSIR migration project to OSS and were verified by experts of the Vula project team. It was a great experience to capture process models comprising of generic migration processes (identified and documented for this type of a project) and to elaborate on generic migration activities that can be replicated by organisations migrating to OSS in future.

5.4.1 (a) Identify the high-level process model

Shown in Figure 5.2 is the repetition of a high-level process model diagram and its set of subprocess models which were refined further (previously described in Chapter 4). This diagram reveals the extracted process reference models for the OSS migration in a high-level hierarchy. These process reference models will be listed and discussed in this chapter and mainly the ones that are acceptable and applicable when it comes to the OSS migration. Process models highlighted in red text are the ones which were found to be generic during the study, the ones in blue text can be generic or non-generic depending on a choice made by the organisation and those in black text are not generic.

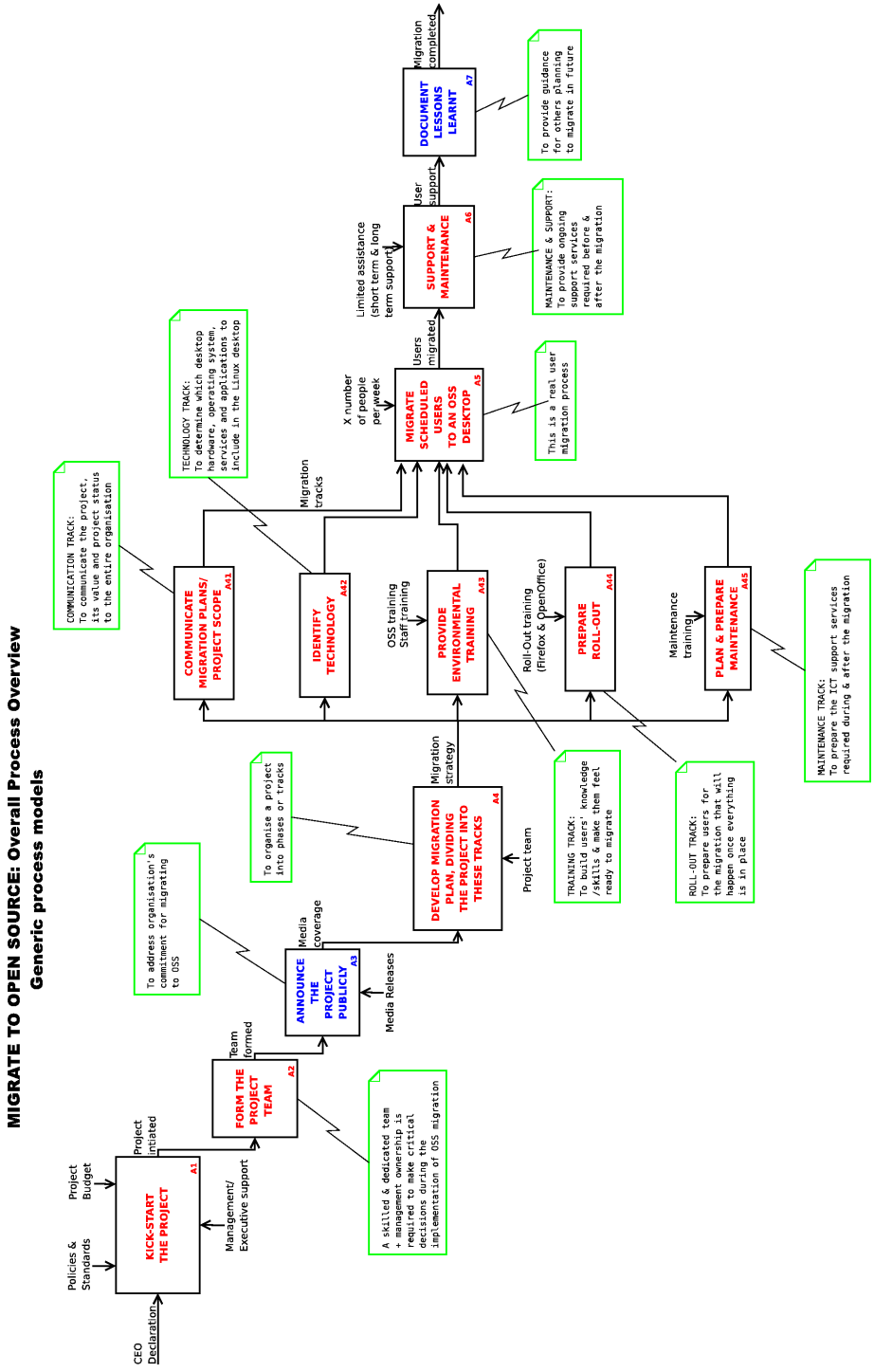

Figure 5.2: Highlighted process reference models in a High-level diagram

5.4.1 (b) List of processes

Following this structure, the list of processes shown in the high-level diagram comprised of a set of 7 processes, where several processes were found to be generic and some not. Here is a list of those 7 processes:

A1= Kick-start the project
A2= Form the project team
A3= Announce the project publicly
A4= Develop migration plan, divide the project into tracks
A5= Migrate scheduled users to an OSS desktop
A6= Support and maintenance
A7= Document lessons learnt.

The next step after the listing of the 7 processes was to determine the possible generic migration processes (process reference models) that are bound to re-occur in other organisations when it comes to OSS migration projects. Table 5.1 below has listed in detail the 7 processes that are generic and non-generic and the reasons why each process was considered generic or not.

Process Name	**Generic (Y/N/C)**	**Motivation**
Kick-start the project (A1)	Y	This activity is generic because it is a process that must take place in order to initiate or introduce the project.
Form the project team (A2)	Y	This activity is generic because a dedicated team is required to plan the migration.
Announce the project publicly (A3)	C	This activity is neither generic nor non-generic because not all projects need to be announced publicly. If you want people to know about this project then you can inform the public about it.
Develop migration plan, divide the project into tracks (A4)	Y	This activity is generic because some projects are too large to be handled by just one person. The migration plan is needed in order to create a platform to run this project from the start until the end and be able to determine what infrastructure or resources are needed to make this project a success.
Communicate migration plans or project scope (A41)	Y	This activity is generic because this is the time when the project team will have to ensure that the main stakeholders and end-users are well-informed about the migration project and are motivated and prepared to move from the current state to the desired one.
Identify technology (A42)	Y	This activity is generic because a plan is required to outline the approach used in selecting technology and designing a standard desktop.
Provide environmental training (A43)	Y	This activity is generic because a plan outlining how training will be implemented is required.

Prepare roll-out (A44)	Y	This activity is generic because this plan will describe the strategy for migrating the majority of users to an OSS desktop and it will also address the support structures that have to be in place in order to enable the roll-out.
Plan and prepare maintenance (A45)	Y	This activity is generic because with a maintenance plan in place, preparations must be done as to how support services will be provided to users currently using the proprietary desktop and to users who have just migrated to OSS.
Migrate scheduled users to an OSS desktop (A5)	Y	This activity is generic because it explains the process of the actual migration of the desktop.
Support and maintenance (short term and long term) (A6)	Y	This activity is generic because even after the migration the current proprietary desktop must still be supported as well as the new OSS desktop environment.
Document lessons learnt (A7)	C	This activity is generic because documenting projects such as this one can be very useful to those organisations and businesses that might consider migrating to an open source environment in future.

Table 5.1: Process reference models for the High-level diagram

Out of these 7 processes which were identified, 5 processes were considered to be generic excluding the *Announce the project publicly (A3)* process and *Document lessons learnt (A7)* process. Unfortunately these two were neglected because they are not crucial processes that every organisation can act upon. That means it is not imperative for every organisation to publicly inform everyone about their projects or share lessons learnt about the work they did. To share information like this is a choice that needs to be made by an organisation. The rest of the 5 processes which were examined to determin if they are generic are distinguished below from those that were considered to be non-generic.

The process *Kick-start the project (A1)* is generic because for every project an announcement has to be made to declare the project open.

Form the project team (A2) process is hard to ignore, because a project needs a proper dedicated team in place as it will not be easy for one person to handle the project on their own. Having extra hands is crucial.

Announce the project publicly (A3) process is not generic due to reasons which were specified above.

Develop migration plan, divide the project into tracks (A4) process is essential and generic because without it there will be no platform created to determine how to run this project from the start until completion. The migration plan is needed in order to provide or divide tasks equally to each member of the team. The projects must be divided into tracks and also be sub-divided into several

tracks or phases. Each track will be led by someone in the team whose passion or day-job is fully aligned with the that type of assignment. Migration plans are also needed to determine the kind of infrastructure or resources needed to make the project a success.

The A5 process which is *Migrate scheduled users to an OSS desktop* is also generic because this stage signifies completion. That means most of the work regarding the migration has been done and all that is left is to reach everyone within the organisation to migrate to OSS.

Support and maintenance A6 process is generic because it entails a short-term and long-term support system plan for the whole organisation.

The *Document lessons learnt (A7)* process can be generic or non-generic. The work done on a specific project can be published if one wants everyone in an organisation to read about it.

After completion of the listing, these generic and non-generic processes of a high-level diagram are then refined to lower-levels (or second-levels) of the above specified high-level process model.

5.4.2 Identify and list the second-level process models

As for the lower-levels process models (or child diagrams in second- level) which consist of the refined processes (or subprocesses) of the high-level process model diagram, the first process to focus on will be *Kick-start the project (A1)*, see Figure 5.3. This process consists of the following set of subprocess models listed below immediately after Figure 5.3; some of them identified as generic and others as non-generic. Note that only subprocesses highlighted in red text were considered to be generic, the ones in blue text rely on choices made by the organisation. Table 5.2 provides various descriptions on why some subprocesses were found to be generic and others not.

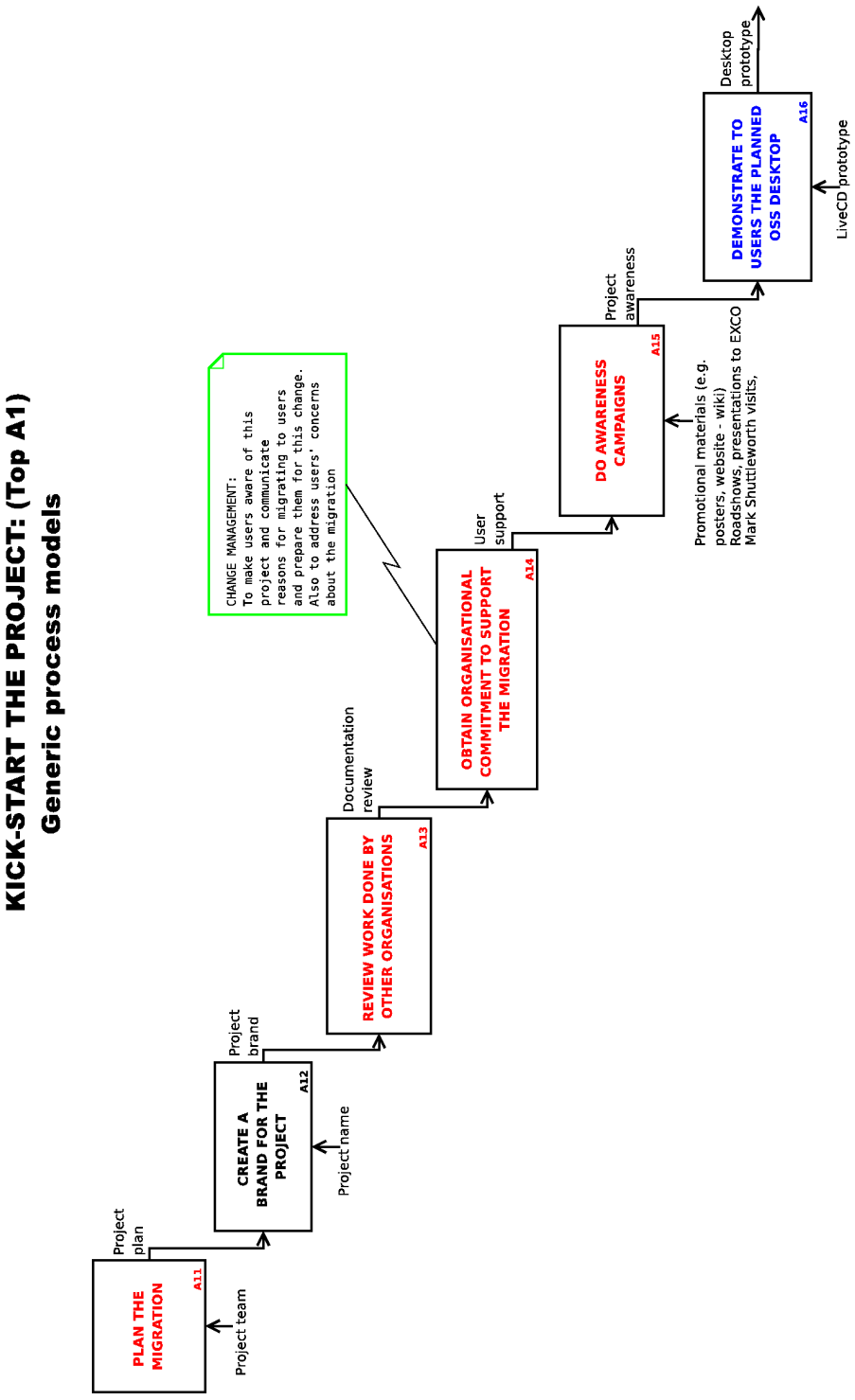

Figure 5.3: Highlighted process reference models for the *Kick-start the project (A1)*

5.4.2 (a) List of subprocesses for *Kick-start the project (A1)*

A11= Plan the migration
A12= Create a brand for the project
A13= Review work done by other organisations
A14= Obtain organisational commitment to support the migration
A15= Do awareness campaigns
A16= Demonstrate to users the planned OSS desktop.

Process Name	**Generic (Y/N/C)**	**Motivation**
Plan the migration (A11)	Y	This activity is generic because a project plan will help to create a platform to run this project from the start until the end and also guide you in identifying some of the pitfalls that can be avoided.
Create a brand for the project (A12)	N	This activity is not generic because some organisations will see no need in naming the project, while others might see it as a good idea as they will be able to use the brand as a logo or an icon on desktops and on published materials.
Review work done by other organisations (A13)	Y	This activity is generic because in order to know what you are letting your organisation into, a detailed investigation has to be done first. And lot of papers needs to be read and it includes looking at what others have done and how they managed to achieve good results.
Obtain organisational commitment to support the migration (A14)	Y	This activity is generic because without the support from users, this can lead to the down fall of the project. So ensuring that everyone is aware of this project will enable you to receive positive support from end-users.
Do awareness campaigns (A15)	Y	This activity is generic because the campaigns will remind and motivate users about the seriousness of this project.
Demonstrate to users the planned OSS desktop (A16)	C	This activity is neither generic nor non-generic because it is not essential to have a prototype desktop ready when coming into consult with users. However, if you do have a prototype that you can show to users then do not hesitate to do so.

Table 5.2: Process reference models for the *Kick-start the project (A1)*

Subprocess (A11) *Plan the migration* is specified as generic because without a plan in place any project might fail, as there will be no guidance or platform to run this project from the start until the end. Therefore, the migration plan is a crucial activity at the beginning of the project.

In terms of the subprocess *Create a brand for the project (A12)*, this activity was found not to be generic because some organisations may see no need in

naming their project, while others might see it as a good idea to use the brand name to advertise themselves when seeking funding from other stakeholders.

With the subprocess *Review work done by other organisations (A13)*, the migrating organisation is required to know what they are letting their organisation into, therefore a detailed investigation has to be done first and lot of papers needs to be read. That includes looking at what others have done and how they have managed to achieve good results. This subprocess is generic.

To *Obtain organisational commitment to support the migration (A14)* it is important to ensure that everyone within the organisation is aware of the project, because without the support from users or the organisation's employees the project could fail. This subprocess is generic. Users are the main drivers of the end product because it will be recieved and used by them at the completion of the project.

Do awareness campaigns (A15). This subprocess is generic but requires proper planning and time, because the campaigns will act as a reminder and motivation to users about the seriousness of this project.

Demonstrate to users the planned OSS desktop (A16). This subprocess is partially generic and non-generic because it is not crucial to have a prototype desktop ready when starting with the project, but should you have a prototype that you can show to users then do not hesitate to do so.

5.4.2 (b) List of subprocesses for the *Form project team (A2)*

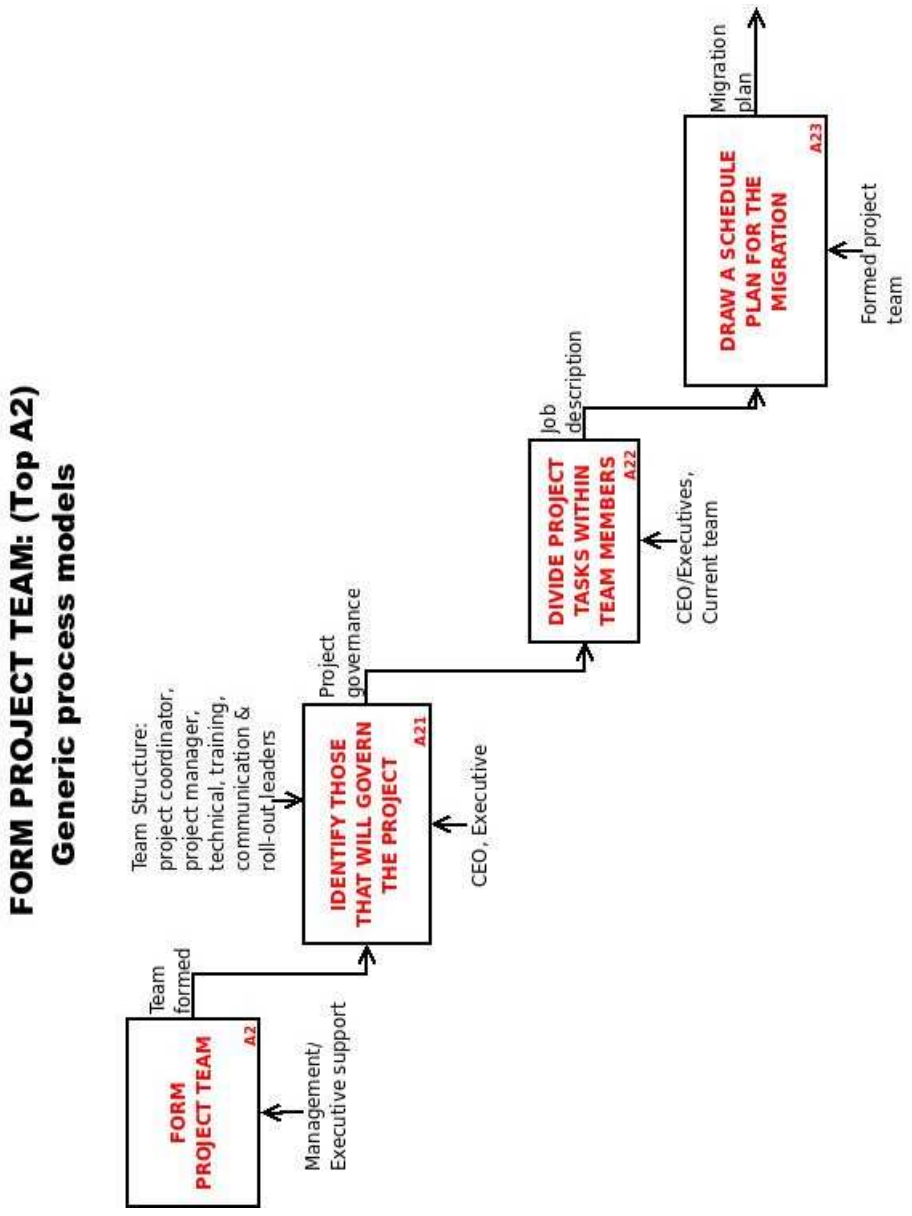

Figure 5.4: Highlighted process reference models for *Form project team (A2)*

A2= Form project team
A21= Identify those that will govern the project
A22= Divide project tasks within team members
A23= Draw a schedule plan for the migration.

Process Name	**Generic (Y/N/C)**	**Motivation**
Form project team (A2)	Y	This activity is generic because a dedicated team with the required skills is worth having to plan the migration. This team will provide solutions to arising problems and ensure that the project runs smoothly.
Identify those that will govern the project (A21)	Y	This activity is generic because knowing who does what within the project will make it easier to divide tasks amongst all the team members. As such it is highly important to know who will lead the project and also the particular tasks members of a team will be responsible for.
Divide project tasks within team members (A22)	Y	This activity is generic because every member in a team will now know the particular tasks they will be responsible for.
Draw a schedule plan for the migration (A23)	Y	This activity is generic because a plan will help to create a platform to run this project from the start until the end. It will also identify some of the pitfalls that can be avoided.

Table 5.3: Process reference models for *Form project team (A2)*

As shown in Figure 5.4 and Table 5.3 all the subprocesses of the *Form project team (A2)* are generic, because the actual migration needs a dedicated team that can plan the migration and ensure that any obstacles are taken care of and that everything runs as smoothly as possible. The team must consist of several members and each member will be responsible for the tasks given to him/her. This was the case for the CSIR migration project, a committed Vula Team was formed to take all the responsibility concerning the migration.

This activity is generic because a team with multi-disciplinary skills is a basic requirement. Most migration projects are too large to be handled by one person, thus assistance from other team members is of importance. Every member in a team must be responsible for particular tasks given to him/her and must complete them successfully. It could be the responsibility of each individual in a team to choose tasks that they feel comfortable with or the CEO or even the project manager can delegate those tasks based on their work experience. Dr Sibisi did this when he appointed Laurens Cloete as the project coordinator, who then identified potential candidates to be part of the team.

5.4.2 (c) List of subprocesses for the *Announce project publicly (A3)*

A31= Announce the project internally and externally
A32= Invite Media.

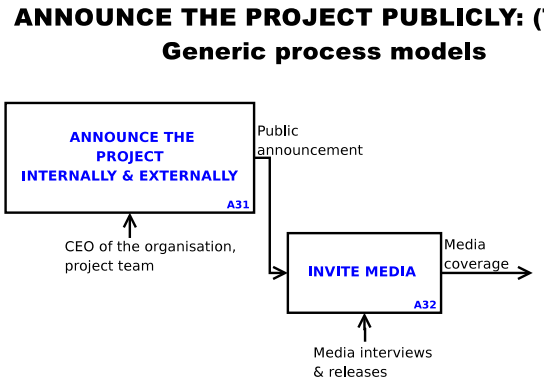

Figure 5.5: Highlighted process reference models for *Announce Project Publicly (A3)*

Process Name	Generic (Y/N/C)	Motivation
Announce the project internally and externally (A31)	C	This activity is neither generic or non-generic because, as mentioned earlier if you want other people to know about this project then you can inform the public about it, if not then you can just keep mum about it.
Invite Media (A32)	C	This activity, which is dependent on the above-mentioned process, is also neither generic nor non-generic because it can only take place once you have chosen to inform the public about the project. You can then invite the media to write a piece about this project.

Table 5.4: Process reference models for *Announce Project Publicly (A3)*

This process *Announce the project publicly (A3)* on its own is partially generic and non-generic, hence the subprocesses were highlighted in blue text for this activity. Furthermore, it is commonly believed that it is not essential to introduce any project to the outside world or motivate as to the reasons that led to the initiation of the project. It can be important for projects like the CSIR migration to OSS to be introduced as part of economical and technological development, however this is not a forceful action. Therefore the two subprocesses that this process *Announce the project publicly (A3)* consist of were regarded as a choice that the organisation or individual must make, depending on who is leading the project.

5.4.2 (d) *Develop migration plan, divide the project into tracks (A4)*

As specified previously in Chapter 4 this process involves the migration tracks: Communication Track, Technology Track, Training Track, Roll-out Track, Maintenance Track. Each of these migration tracks were analysed on its own, with a number of sub-subprocesses (or atomic processes) emerging from each. Listed and discussed below are those tracks' sub-subprocesses (or atomic processes).

(i) List of sub-subprocesses for the Communication Track

In contrary to the abovementioned process models which were only refined in the second-level, most of the following process models for the Communication Track were decomposed further into third-level as shown in the list of sub-subprocesses (or atomic processes) below immediately after Figure 5.6. In

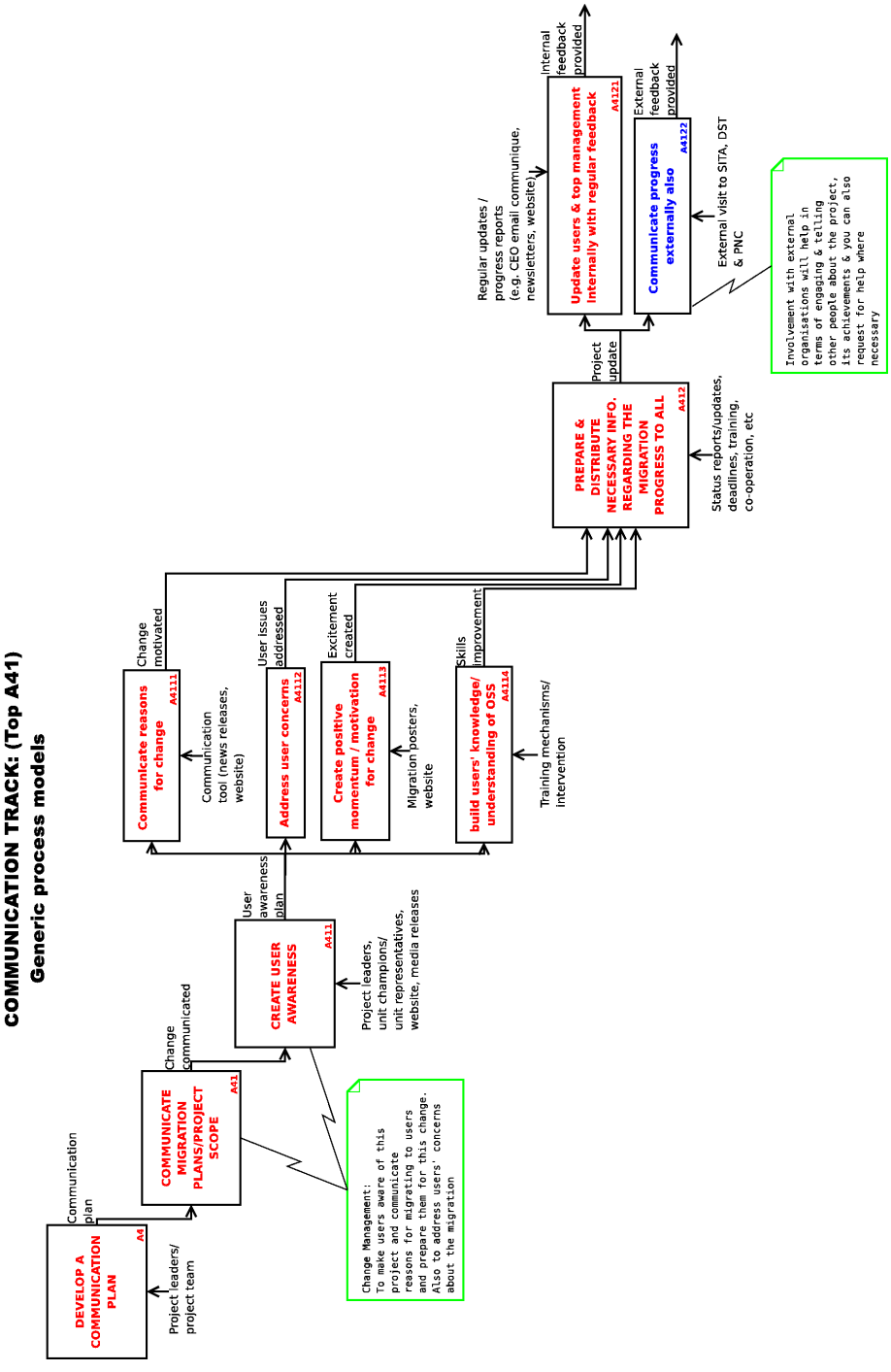

Figure 5.6: Highlighted process reference models for Communication Track

A4= Develop a communication plan
A41= Communicate migration plans or project scope
A411= Create user awareness
A4111= Communicate reasons for change
A4112= Address users' concerns
A4113= Create positive momentum or motivation for change
A4114= Build users' knowledge or understanding of OSS
A412= Prepare and distribute necessary information regarding the migration progress to all
A4121= Update users and top management internally with regular feedback
A4122= Communicate progress externally also.

Here is a brief description of why some of the Communication Track sub-subprocesses (or atomic processes) were found to be generic and others partially generic. Note that only processes highlighted in red text were considered to be generic, the ones in blue text relied on choices made by the organisation.

Process Name	**Generic (Y/N/C)**	**Motivation**
Develop a communication plan (A4)	Y	This activity is generic because a plan is required to outline the approach that will be followed to communicate the project to users.
Communicate migration plans or project scope (A41)	Y	This activity is generic because this is the time when the project team will have to ensure that the main stakeholders 'end-users' are well-informed about the migration project and are motivated and prepared to move from the current state to the desired one.
Create user awareness (A411)	Y	This activity is generic because besides making users aware of the project, the team should use this particular time to achieve buy-in across the organisation and ensure that the organisational support and commitment is gained.
Communicate reasons for change (A4111)	Y	This activity is generic because motivating and persuading users within the organisation to support the migration and making them realise the good investment they are making by moving to an open source environment is a good idea.
Address users' concerns (A4112)	Y	This activity is generic because this is the chance that users will have to raise their concerns about moving to another distribution. Here the team will have to assure end-users that their needs will be met where possible. No one is willing to put themselves and their work at risk. If other issues raised by end-users cannot be resolved, talk to them or negotiate with them.
Create positive momentum or motivation for change (A4113)	Y	This activity is generic because once users have been reassured that their needs will be met, positive momentum will continue to increase for the migration. This kind of awareness will always stick in users' minds and most of them will never forget this initiative.

Build user knowledge or understanding of OSS (A4114)	Y	This activity is generic because users who are clueless about the new open source environment will be exposed to it and learn more about it through reading and training.
Prepare and distribute necessary information regarding migration progress to all (A412)	Y	This activity is generic because the entire organisation and everyone that will affected by the migration will always be informed or updated about the status of the migration.
Update users and top management internally with regular feedback (A4121)	Y	This activity is generic because everyone within your organisation will be kept informed about the migration and of any changes that might be taking place.
Communicate progress externally also (A4122)	C	This activity is neither generic nor non-generic because depending on the organisation's principles it is not essential to update everyone outside the organisation about this project. If the external entities that the organisation collaborates with will be affected by this change, you must let them know of the challenges faced by the organisation with regards to the migration.

Table 5.5: Process reference models for Communication Track

The Communication Track's focus is on developing a communication plan. Developing such a plan takes time and effort, thus this activity is considered generic because it is crucial to develop a communication plan that fully supports the migration to OSS. This plan must outline the communication approach that must be followed, including the major communication venues to be used during the project for meetings and so forth.

Communication plans can be used to make the migration process and progress visible to everyone who will be affected and this is where *Communicating migration plans or project scope (A41)* comes into place as depicted in Table 5.5. The purpose of the communication plan is to allow the main stakeholders and users to be informed about the migration project and to be motivated and prepared to move from the current state to the desired one. The *Communicate migration plans or project scope (A41)* process was further refined into the third-level process models which were mainly found to be generic as analysed in Table 5.5.

The next sub-subprocess *Create user awareness (A411)* is generic because it is one of those steps that must be taken in order to make users aware of the project. The reasons that motivated this kind of change to take place must also be prepared and communicated to the corncerned stakeholders; in this case the CSIR's employees. Therefore the generic sub-subprocess is *Communicate reasons for change (A412).*

The next generic sub-subprocess *Address users' concerns (A413)* suggests pro-

viding the corncerned parties, that is users, with a chance to raise their concerns about the movement to another distribution. Remember that not everyone within the organisation will be happy about this kind of change, therefore it is best to first listen to users' aggreviations before continuing further with the move and reach a compromise where possible.

The sub-subprocess *Create positive momentum or motivation for change (A414)* is also generic. Once users are reassured that their needs will be met, excitement generated for the project sticks in users' minds and most of them never forget about this initiative and continue to offer their support. At the same time it must be noted that not everyone will support the migration, therefore it is important to train all of those who don't know this new environment. To get users to know or understand how to operate the new distribution the sub-subprocess (A415) *Build users' knowledge or understanding of OSS* is relevant. Then everyone within the organisation must be kept up to date with all the necessary information regarding the migration, therefore sub-subprocess *Prepare and distribute necessary information regarding migration progress to all (A416)* and sub-subprocess *Update users and top management internally with regular feedback (A417)* are generic. As for reporting progress outside the organisation, it is a choice that must be made by organisational leaders based on business needs and collaborations with its stakeholders.

(ii) List of sub-subprocesses for the Technology Track

The same procedures applied in the Communication Track were also used for the Technology Track, where the focus was not only based on second-level process models, but also on the third and fourth-level as shown in Figure 5.7. However, due to limited space only the Technology Track's sub-subprocesses on third and fourth-level are analysed in Table 5.6, the rest of the details can be viewed further in documentation provided on the CD.

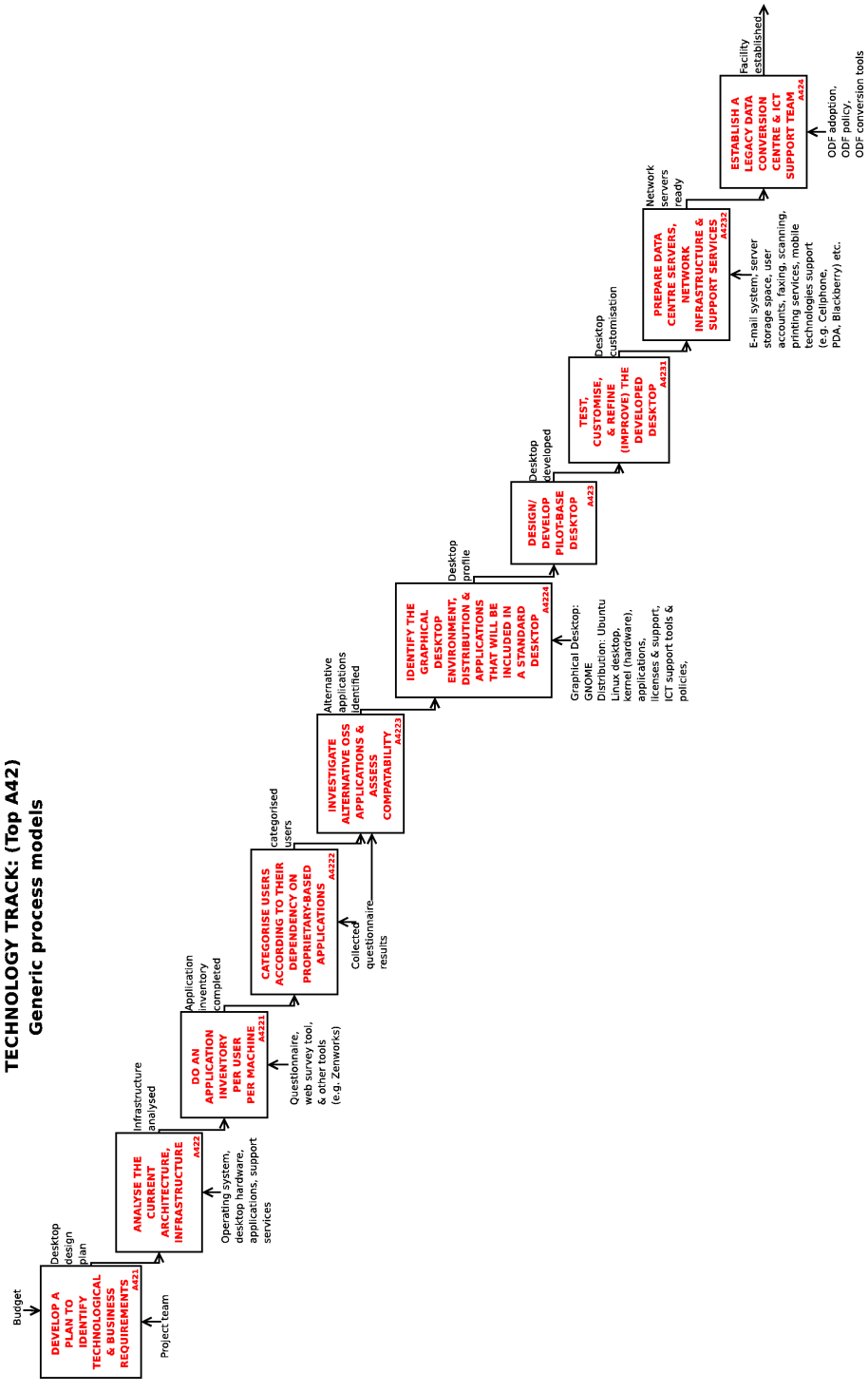

Figure 5.7: Highlighted process reference models for Technology Track

A421= Develop a plan to identify technological and business requirements
A422= Analyse the current architecture and infrastructure
A4221= Do an application inventory per user per machine
A4222= Categorise users according to their dependence on proprietary-based applications
A4223= Investigate alternative OSS applications and assess compatability
A4224= Identify the graphical desktop, environment, distribution and applications that will be included in a standard desktop
A423= Design or develop a pilot-base desktop
A4231= Test, customise and refine the developed desktop
A4232= Prepare data centre servers, network infrastucture and support services
A424= Establish a legacy data conversion centre and ICT support team.

Process Name	**Generic (Y/N/C)**	**Motivation**
Develop a plan to identify technological and business requirements (A421)	Y	This activity is generic because a plan is required to outline the approach used in selecting technology, designing a standard desktop, putting in place the architectural or network infrastructure and other technical aspects or developments that may be needed for the migration.
Analyse the current architecture and infrastructure (A422)	Y	This activity is generic because it will help you to understand the current proprietary desktop and to determine which operating system, desktop hardware, applications, and services closely match the open source desktop environment.
Do an application inventory per user per machine (A4221)	Y	This activity is generic because it will help to identify the currently used applications on the existing proprietary desktop, in order to know what kind of technological and business applications must be included in the open source desktop environment to match the current environment.
Categorise users according to their dependence on proprietary-based applications (A4222)	Y	This activity is generic because it will help the project team to easily identify less-risk users who can be migrated first and have a plan in place for individuals who will be running critical, in-house or special proprietary-based applications.
Investigate alternative OSS applications and assess compatability (A4223)	Y	This activity is generic because once the alternative applications and their functionality are compatible with the current ones used on the proprietary desktop, then it is easier to determine the matching applications that must be included on the open source desktop.

Identify the graphical desktop, environment, distribution and applications that will be included in a standard desktop (A4224)	Y	This activity is generic because here you will be making a final decision about which distribution and associated applications best support your business objectives in terms of interoperability, certification, product support, ease of installation, software management, and open source innovation. The above results will determine which route to take.
Design or develop a pilot-base desktop (A423)	Y	This activity is generic because this is the desktop that will closely match the existing proprietary desktop and will contain every OSS-related application that is critical to every user's work to enable them to continue with their work without any disruptions.
Test, customise and refine the developed desktop (A4231)	Y	This activity is generic because continuous tests or improvements will ensure that the developed open source desktop functions well and meets users' requirements.
Prepare data centre servers, network infrastucture and support services (A4232)	Y	This activity is generic because once the open source desktop has been developed, e-mail server, temporary backup facility for data migration and network infrastructure also need to be ready to support open source desktops. This is to ensure that desktops connect to servers, storage, printers, and other network devices. Everything must operate smoothly.
Establish a legacy data conversion centre and ICT support team (A424)	Y	This activity is generic because the adoption of an ODF standard means that all documents, spreadsheets and presentations needs to be converted to an OpenOffice document format (ODF) and tested by a small conversion support team to ensure that all documents meet an open document standard.

Table 5.6: Process reference models for Technology Track

From the above evaluation of the Technology Track, there are many technical aspects to be considered when planning an OSS desktop migration. Therefore a plan for the Technology Track is essential in the initial stage, as it is required to address things like the design of the standard desktop, architectural and back-end implications of the migration and also to outline the approach used for selecting technology and to cover any development of applications that may be required. Hence we declare the sub-subprocess *(A421) Develop a plan to identify technological and business requirements* as generic.

In this regard after doing all the planning, the analysis of the current desktop is required to determine which operating system, desktop hardware, applications and services will be included in the Linux desktop environment and evaluate how close a match the new environment is to the existing Microsoft Windows desktop. It is also recommended that first a study is done of what other companies have done to have successful migrations and that their work be used to guide you in planning the migration for your organisation. For these reasons and more which are provided in the documentation on the CD this sub-subprocess *(A422)*

Analyse the current architecture and infrastructure was found to be generic.

Following the planning and assessment is the sub-subprocess *Do an application inventory per user per machine (A4221)*, it involves completing an application inventory for the current desktops. This sub-subprocess is highly recommended as it will help to identify and classify desktop services and applications. This basic set of desktop services and applications include amongst others an office or productivity suite, faxing, file, scanning and printing services, e-mail, calendar, virus scanning and spam filtering, POP, IMAP, DMS, instant messaging, audio or video playback, browser, etc. At the CSIR a questionnaire was used to gather information of applications utilised by users at the time. In some instances a program can be written that will enable one to search and extract the information from the user's desktop applications.

After completing an application inventory of the current desktops, it is also recommended that users must be categorised according to their dependence on Microsoft Windows-based applications; sub-subprocess *(A4222) Categorise users according to their dependence on proprietary-based applications.* This categorisation will be based on a collected inventory of desktop applications done in the previous processes. This will help the project team to easily identify the less-risk users who can be migrated first.

As part of planning a desktop migration, it is important to determine if the OSS environment (Linux) supports the applications that are comparable to the currently used ones on the Microsoft Windows desktop. Then it is easier to determine the matching applications that must be included on the Linux desktop. Sub-subprocess *(A4223) Investigate alternative OSS applications and assess compatability.*

The following sub-subprocess *Identify the graphical desktop, environment, distribution and applications that will be included in a standard desktop (A4224)*, was considered generic and critical because it involves making that final decision about which distribution and its associated applications best support the business objectives in terms of interoperability, certification, product support, ease of installation, software management and open source innovation. This is a critical point which will determine the design of the future desktop that will cater for the organisation's needs and it will be expected to meet users' needs in the same way that the previous Microsoft Windows desktop used to.

To design an OSS desktop takes times and effort, therefore it is important to first determine which desktop hardware, operating system, services and applications will be included on the OSS desktop profile and how close a match that profile must be to the existing Microsoft Windows desktop. Every application that is most relevant to users' work must be included on this desktop. That includes OSS alternatives of applications that are found within the Microsoft Windows environment. Sub-subprocess *(A423) Design or develop a pilot-base desktop.*

Once the desktop has been designed and put together, continuous tests and improvements must be performed to ensure that the developed Linux desktop

is well-functioning. A lot of time must be spent on developing this desktop as users want a 100 percent working desktop that will enable them to do their work without any disruptions Patience and cautiousness are highly recommended in this instance. Process *(A4231) Test, customise and refine the developed desktop.*

Once the OSS desktop has been developed and continuously tested, additional factors such as an e-mail server, temporary backup facility for data migration, network infrastructure may also need to be considered. These services must be prepared to support OSS desktops, that is to ensure that desktops are constantly connected and communicating with servers, storage, printers and other network devices. Backup systems must be kept up to date to ensure that users' data is stored and kept safe at all times and there will be a backup in case something goes wrong. Without these kinds of resources in place, the migration on the technological side may be doomed. Process *(A4232) Prepare data centre servers, network infrastucture and support services.*

Remember that migrating to an OSS environment involves the adoption of an OpenOffice document format (ODF). This means that all documents, spreadsheets and presentations must support this open standard by being converted to a standard format of OpenOffice and tested by a small conversion support team to ensure that all documents meet open document standards. Sub-subprocess *(A424) Establish a legacy data conversion centre and ICT support team.*

The processes of the Technology Track, described above, were all considered to be generic as shown in Figure 5.7 and Table 5.6, because they were all identified as critical steps required to be taken by skilled migration project team members, when they design and develop an OSS desktop system that will suit users' needs within the organisation.

As it was specified in the previous chapter the sub-subprocess *(A4231) Test, customise and refine the developed desktop*, was also refined further into a set of atomic processes. From this set of atomic processes, the generic ones were identified and can be viewed in Table 5.7 and also in the documentation provided on the CD with more details.

Process Name	Generic (Y/N/C)	Motivation
Perform testing (A42311)	Y	This activity is generic because performing tests will ensure that everything runs smoothly. This is a vital process that will determine whether the developed desktop is ready for use. If not, more tests have to be conducted until everything works as required. If the developed desktop works perfectly it can be rolled out into a production environment.
Set-up a test lab (A42312)	Y	This activity is generic because the proposed open source desktop and its associated applications must be tested in a development lab, or in a pilot or limited production setting before rolling it out into a production environment.
Test the operating system (desktop) and associated applications (software, hardware, legacy systems compatability) (A42313)	Y	This activity is generic because these kinds of tests will ensure that the developed desktop and its associated applications work as required and also ensure that they connect or communicate with the systems used within the organisation such as e-Procurement, ERP, HR, Workflow systems etc.
Conduct pilot tests (A42314)	Y	This activity is generic because doing these kinds of tests will help to identify potential areas such as those that would need to be covered in a user training manual or that the help desk and support staff will need to be aware of. Therefore, the training and instructions manual prepared for users and technicians can be updated as needed during testing.
Ensure functionality and efficiency of the whole system (A42315)	Y	This activity is generic because with everything working as expected, it means the entire organisation will be faced with no or little disruption when it comes to doing their work.

Table 5.7: Process reference models for *Testing and customising the desktop (A4231)*

This sub-subprocess briefly articulates how the designed desktop is improved, to ensure the functionality of the whole OSS desktop system. Once the initial stage of planning and assessment have been done, the proposed OSS desktop and its associated applications must be tested in a development lab or in a pilot-limited production setting before rolling it out into a production environment. Doing these kinds of tests can help identify potential areas that the project team will need to be aware of before distributing the OSS desktop to the entire organisation. In this instance, a group of people (or real users) will be used for pilots. This group of real users must include those that are familiar with the OSS desktop and even some that are not familiar with the new desktop. They will all undergo a training session whereby they will try out the new OSS desktop, analyse the system carefully and provide feedback that must be taken seriously and handled by the project team to provide solutions.

(iii) List of sub-subprocesses for the Training Track

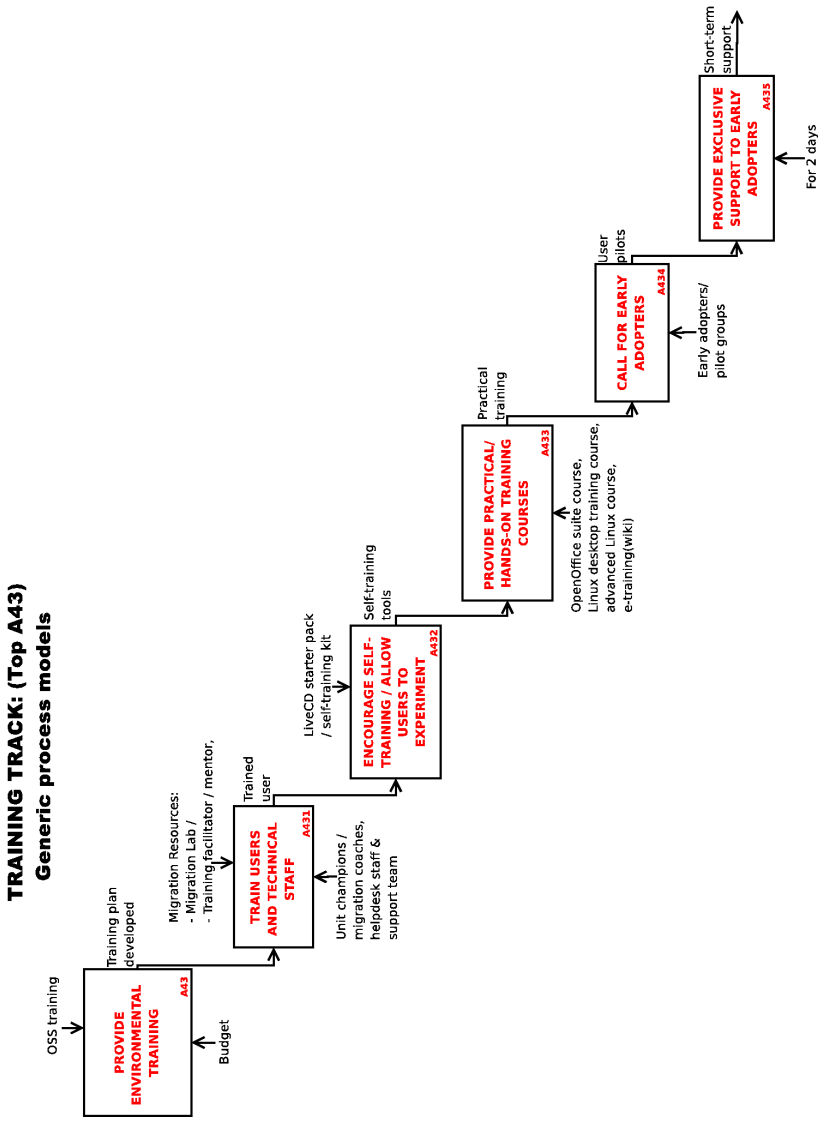

Figure 5.8: Highlighted process reference models for Training Track

A43= Provide environmental training
A431= Train users and technical staff
A432= Encourage self-training or allow users to experiment
A433= Provide practical or hands-on training
A434= Call for early adopters
A435= Provide exclusive support to early adopters.

Process Name	Generic (Y/N/C)	Motivation
Provide environmental training (A43)	Y	This activity is generic because a plan outlining how training will be implemented is required. A budget must be in place to do this task, as this is one of the activities that will be cost-effective to the organisation. The time factor must also be considered, as users will undergo a 2-day training session so it is important to have a strategy in place to ensure that this will not affect business productivity in any way.
Train users and technical staff (A431)	Y	This activity is generic because it will help users to familiarise themselves with the new desktop, the same applies to the support services team. The support team must be able to relate to or understand calls for assistance regarding open source and resolve such problems.
Encourage self-training or allow users to experiment (A432)	Y	This activity is generic because equipping users with the necessary material such as the LiveCD starter-pack / kit will enable them to experiment or teach themselves about the new open source desktop. At the same time they will be able to install applications on their own, but if they struggle the support team must always be prepared to provide them with help.
Provide practical or hands-on training (A433)	Y	This activity is generic because this type of training will allow users to physically use the machine/desktop in order to learn and understand all the things that are critical to enable them to perform their jobs whether it is in OpenOffice or on the Internet. What matters is that they must be able to do their work without any disruptions.
Call for early adopters (A434)	Y	This activity is generic because the more users you have volunteering to migrate, the better the migration for the rest of the organisation will go.
Provide exclusive support to early adopters (A435)	Y	This activity is generic because once the early adopters have been migrated and trained, short-term support must be provided to them. This is to ensure that they are on track with the open source desktop and being helped by the support team with any difficulties they come across.

Table 5.8: Process reference models for Training Track

Providing training (A43) is an essential part of the track, thus this subprocess is generic because it involves having the training plan in place, outlining how training of users, ICT technical staff and general training on OSS will be implemented. However, it is important to first find out if there is a budget set aside for training, as this is one of the phases that will be of cost to the organisation. Also time must be considered, if users must undergo 2-day training. A strategy must be in place to ensure that this will not affect business productivity

in any way.

Train users and technical staff (A431). This sub-subprocess is generic because it will help users to familiarise themselves with the new OSS desktop. The same applies to the support services team. In order to provide user support, the support services team (technicians, help-desk staff) must be certified or have the required certification to perform OSS-related problems experienced by users. Therefore, technicians must also undergo a 2-day training course to be trained on an advanced level of the OSS (Linux) in order to understand everything about open standards and OSS. They must know how OSS differs from Microsoft Windows, be able to relate to and understand calls for assistance regarding OSS and resolve such problems.

Encourage self-training or allow users to experiment (A432). This activity is considered generic because equipping users with the necessary skills such as, allowing them to install applications on their own and providing them with help in areas where they could be struggling, will enhance users' knowledge of OSS. This means providing users with all the material they need to teach themselves about the new OSS desktop such as LiveCD starter-pack or kit. However, users must undergo formal training first and after being trained they will be given the material.

Provide practical or hands-on training (A433). By practical training we mean allowing users to physically use the machines (or desktops) in order to learn all the things that are critical to them when it comes to performing their jobs. These users must know everything in detail so that when they go back to their offices they are capable of conducting what they had learnt during the training session.

Call for early adopters (A434). This sub-subprocess is generic because the more users volunteer to migrate, the better the chance of having the rest of the organisation migrating to OSS too. However, ensure that even after the migration the early adopters get assistance from the support team and deployed migration coaches, should they experience any difficulties with the desktop, hence the process *Provide exclusive support to early adopters (A435).*

(iv) List of sub-subprocesses for the Roll-out Track

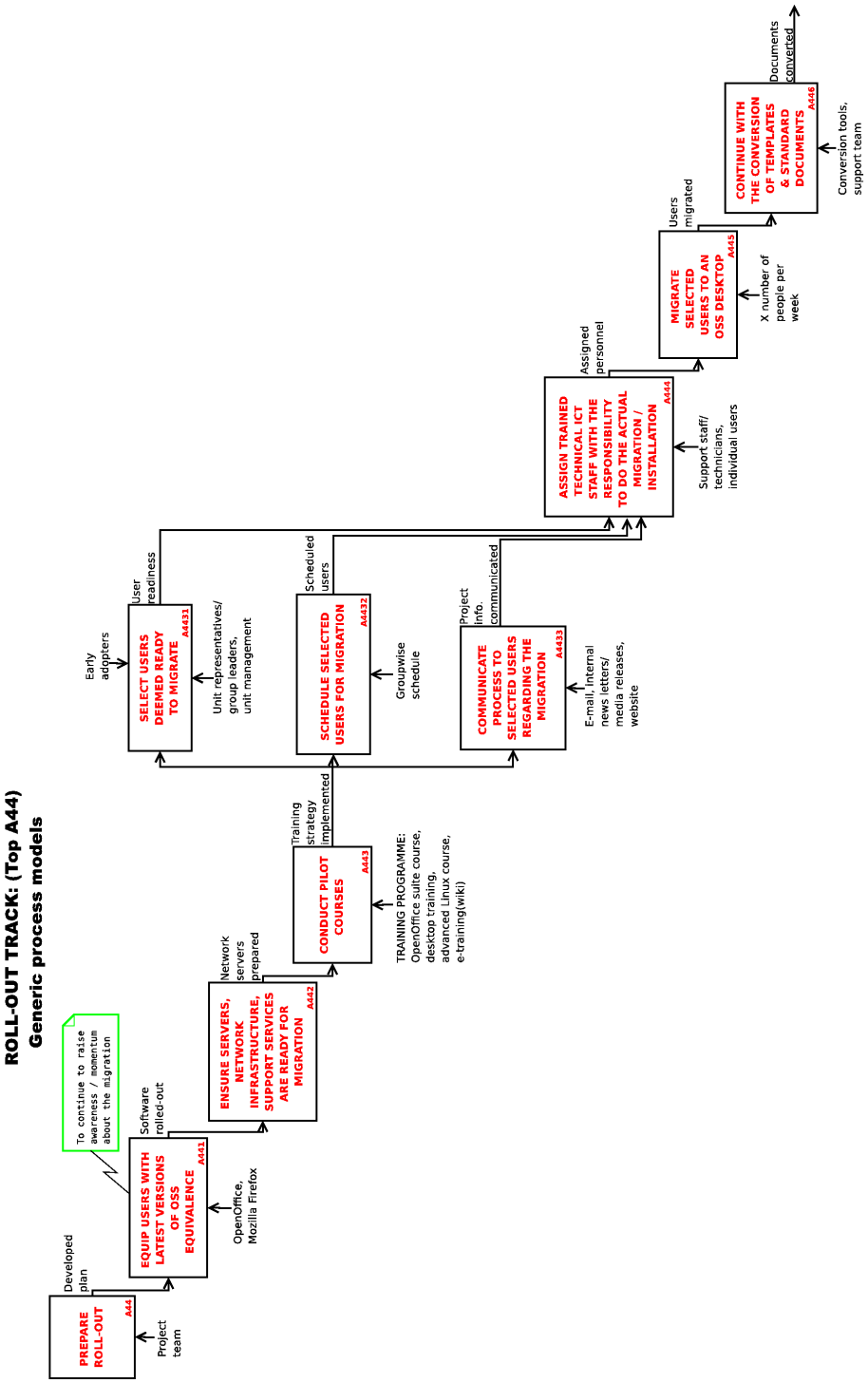

Figure 5.9: Highlighted process reference models for Roll-out Track

A44= Prepare roll-out
A441= Equip users with latest versions of OSS equivalence
A442= Ensure servers, network infrastructure and support services are ready for migration
A443= Conduct pilot-courses
A4431= Select users deemed ready to migrate
A4432= Schedule selected users for migration
A4433= Communicate process to selected users regarding the migration
A444= Assign trained technical ICT staff with the responsibility to do the actual migration or installation
A445= Migrate scheduled users to an OSS desktop
A446= Continue with the conversion of templates and standard documents.

Process Name	**Generic (Y/N/C)**	**Motivation**
Prepare roll-out (A44)	Y	This activity is generic because this plan will describe the strategy for migrating the majority of users to an open source desktop and it will also address the support structures that have to be in place in order to enable the roll-out (e.g. how users will be prepared for the open source migration, how back-end systems such as servers, HR, ERP systems etc. will be upgraded to enable the migration).
Equip users with latest versions of OSS equivalence (A441)	Y	This activity is generic because it will encourage users to use Open source applications like OpenOffice instead of Microsoft Office and get used to its look and feel.
Ensure servers, network infrastructure and support services are ready for migration (A442)	Y	This activity is generic because once the open source desktop has been developed e-mail servers, backup facilities and network infrastructure must all be ready to support open source desktops, to ensure that desktops connect to servers, storage, printers, and other network devices.
Conduct pilot-courses (A443)	Y	This activity is generic because putting a good training programme in place and communicating properly all the necessary information to users will prepare them for this type of change.
Select users deemed ready to migrate (A4431)	Y	This activity is generic because it gives management within each unit (including unit representatives or group leaders) the responsibility to nominate or select users who are deemed ready for migration and encourage and support them to undergo pilot training. Bear in mind that some users will volunteer as early adopters, so the more volunteers you have for each training session, the more the number of migrated users will increase.

Schedule selected users for migration (A4432)	Y	This activity is generic because it prepares users for the real deal. Set up appointments for each individual on Groupwise for when the desktop will be collected, the date and time for the training and the eventual date for the delivery of the migrated desktop.
Communicate process to selected users regarding the migration (A4433)	Y	This activity is generic because its purpose is to ensure that users who have been selected by their respective unit managers for migration, understand the whole process of the migration and the relevant information that they might need to complete the migration. This information will be discussed with them or given to them (such as the time and area where the training session will be taking place and so forth).
Assign trained technical ICT staff with the responsibility to do the actual migration or installation (A444)	Y	This activity is generic because who will be responsible for installations is a critical decision. Some users will prefer to migrate themselves.
Migrate scheduled users to an OSS desktop (A445)	Y	This activity is generic because during this activity the proprietary desktops will be migrated to the new open source desktop while the user is being trained to use the new operating system productively.
Continue with the conversion of templates and standard documents (A446)	Y	This activity is generic because the adoption of a new document format standard indicates that a large number of documents and templates must be converted and be used regularly within the organisation, to allow users time to get used to a new format.

Table 5.9: Process reference models for Roll-out Track

The first aspect to focus on in this regard, is that of planning the strategy for migrating the majority of users to an OSS desktop. The plan must also address the support structures that have to be in place in order to enable this migration and its installations. This plan must be like a roadmap showing how users will be prepared for the OSS migration, how other systems such as servers, HR, ERP systems etc. will be upgraded to enable the migration and also how users will be supported during and after the migration. Without this kind of structure in your plan, it will be a bit difficult to satisfy all the requirements of users during the migration.

Due to adoption of ODF it is essential and highly recommended to equip users within the organisation with the latest versions of the OSS such as OpenOffice and Mozilla Firefox. That way they will be able to experiment with the readily available applications installed on their machines. Therefore, this subsubprocess is considered generic because it will encourage users to use OpenOffice instead of Microsoft Windows Office and so get used to its look and feel. The same applies to leaving Internet Explorer and starting to use Mozilla Firefox.

Once the OSS desktop has been developed, it is essential to keep on track with e-mail servers, backup facilities and network infrastructure to ensure that all the resources are ready to support OSS desktops. That means desktops must connect to servers, storage, printers and other network devices, interoperate with other desktop operating systems as well as directory and authentication services.

Conducting pilots can be a dreadful process, but will definitely allow users to easily navigate their way on the new desktop. Thereafter the users will give feedback regarding the new look and feel of the desktop and this feedback will help the project team to rectify any kind of mistakes that could have been identified by users during the pilot session. This activity is found to be generic. Putting a good training programme in place and a good presenter to communicate properly all the necessary information to users will prepare them for this type of a change. It is recommended to have 1 to 2 days training focusing on OSS and OSS desktop courses. At the same time the team must ensure that there is a place where training will be taking place (be it a lab or any area that is big enough to accommodate the users scheduled for training) and that it is fully equipped with all the required resources such as desktops set-up for training.

When doing pilots, it is recommended to take a small number of users at a time. It is better not to force users to go for this kind of training, but to allow them to volunteer. This sub-subprocess is generic because it gives management within each unit (including unit representatives or group leaders) the responsibility to select users who are deemed ready for migration and to then encourage and support them to undergo pilot training.

Once management has selected individuals deemed ready for migration, an appointment is set up with each individual on Groupwise for the collection of the desktop, the training and eventual delivery of the migrated desktop. The intention is to focus on commitment from management as well as the individual ready to migrate. This stage is considered crucial for the success of this migration.

It must be ensured that the relevant users scheduled for migration, who are the main stakeholders in this regard, and management are aware of all the project information they need to know to prepare them for the migration. Ensure that they know when they are scheduled for training, how their cooperation will be needed and when will they receive their migrated desktops. Once the selected users know this information, this communication channel can be used to update other users, executives and everyone else about progress regarding the migration.

In this sub-subprocess, *(A444) Assign trained technical ICT staff with the responsibility to do the actual migration or installation*, lies the critical decision of who will be responsible for installations. Some users prefer to migrate themselves, while in other cases it is the responsibility of the support staff to do the installations. However, the most important factor is to ensure that the involved ICT support staff is properly trained to do the installations and to resolve any technical-related issues that might be experienced by users. Additionally users can also be trained to do their own installation. Whoever does the migration

should do so with care to ensure no data loss during the migration.

When migrating scheduled users to an OSS desktop, the responsibility to do the installations will be that of ICT support staff. During this sub-subprocess *(A445) Migrate scheduled users to an OSS desktop*, a certain number of Microsoft Windows desktops will be migrated to the new OSS desktop while the user is being trained to use the new OSS operating system productively.

When the CSIR adopted ODF, Dr Sibisi who is the CSIR President/CEO presented talks on ODF and the adoption of the ODF policy, announcing that all the documentation of the organisation must adhere to this new standard document format. This activity is considered generic because the adoption of a new document format standard means that a large number of documents and templates must be converted. Users must be allowed time to get used to a new format.

The next track, Maintenance, is simply about the preparation of putting together the maintenance plan and it reveals how support services will be provided to the entire organisation. This is followed by the sub-subprocess of *Scheduling Users for Migration (A4432)* and after that the process of *Migrating scheduled users to an OSS desktop (A5)*.

(v) List of sub-subprocesses for the Maintenance Track

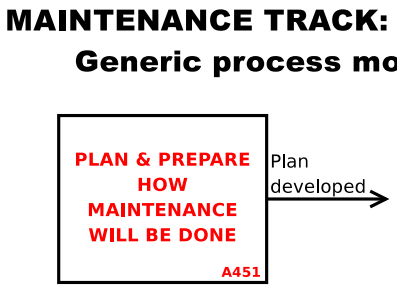

Figure 5.10: Highlighted process reference model for Maintenance Track

A451= Plan and prepare how maintenance will be done.

Process Name	Generic (Y/N/C)	Motivation

Plan and prepare how maintenance will be done (A451)	Y	This activity is generic because with a maintenance plan in place, preparations must be done for how the support services will be provided. Support must be provided to the users currently using the proprietary desktop, but also to users who have just migrated and are now using the open source desktop. Users' problems must be resolved without a lot of delays.

Table 5.10: Process reference models for Maintenance Track

Once a number of users have been migrated, a plan must be put into place as to how the users who are now utilising a different desktop (OSS environment) must be supported while still continuing with the support of the current Microsoft Windows desktop users. With a plan in place, support services can be provided to both the users currently using the Microsoft Windows desktop and to users who have just migrated and are now using the OSS desktop. This plan will ensure that all users' problems are resolved without a lot of delays.

5.4.2 (e) List of sub-subprocesses for *Scheduling selected users for migration (A4432)*

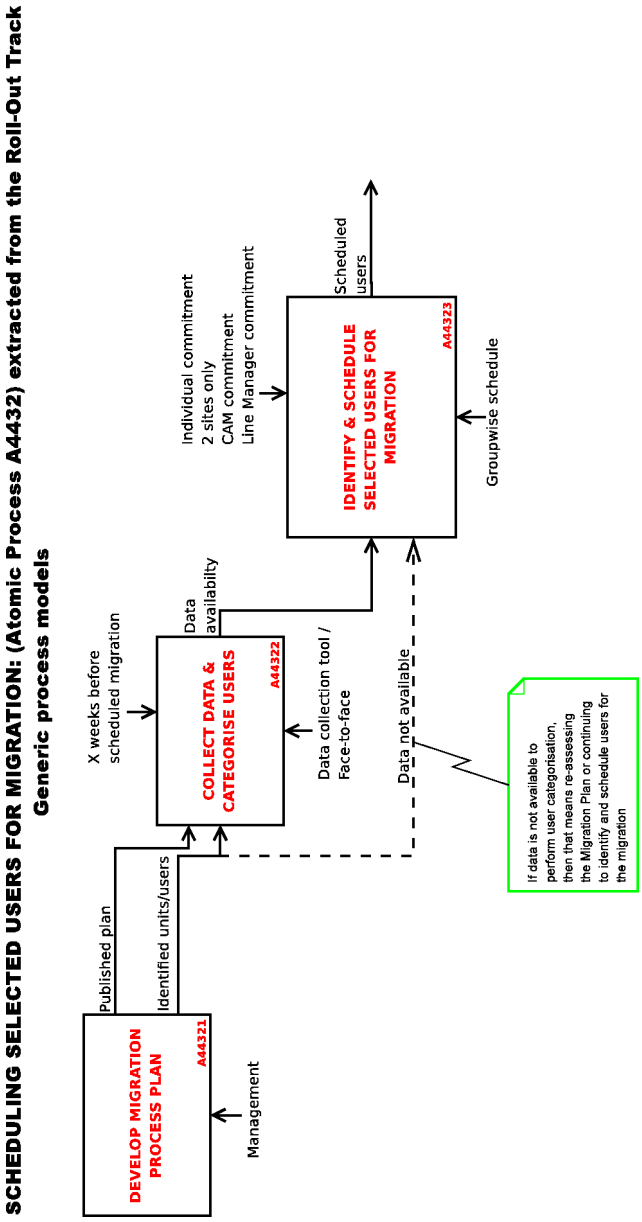

Figure 5.11: Highlighted process reference models for *Scheduling users for migration (A4432)*

A44321= Develop migration process
A44322= Data collection and user categorisation
A44323= Identify and schedule individuals for migration.

Process Name	**Generic (Y/N/C)**	**Motivation**
Develop migration process (A44321)	Y	This activity is generic because this plan will explain the exact migration process, that is, how users will be trained and thereafter get migrated from a current desktop to the open source desktop.
Data collection and user categorisation (A44322)	Y	This activity is generic because it is one of the vital processes that will determine whether the user is ready to be migrated or not. In order to consider a user for migration, a trained coach will first check the list of applications currently used by the specific user, then thereafter make a decision as to whether this user is fit for migration or not.
Identify and schedule individuals for migration (A44323)	Y	This activity is generic because once the above activity has been completed and a decision is made, those users that have been selected for migration will undergo a 2-day training course while their machines are being upgraded or loaded with the open source desktop.

Table 5.11: Process reference models for *Scheduling selected users for migration (A4432)*

The plan for this process consists of the exact migration process scenario for how users will be trained and thereafter get migrated from the current desktop to an OSS desktop. At the CSIR the plan was performed in the following manner: initially the Vula Leadership team presented and explained the migration process to the Executives (OPCO) and after that the responsibility resided with Unit Leadership to populate the 40 migration slots per week, with the users which they elected fit for migration. CSIR Leadership determined the sequence in which their respective Units were migrated within known constraints and guidelines.

A few weeks prior to the scheduled migration, user data was collected by means of a collection tool or face to face interviews. User data refers to the check-list of applications that was handled by migration coaches. The coaches were trained to determine whether a user can be migrated or not and they stated reasons for their decisions. Information relating to applications used by the user was applied to place them into categories and these catogories were migrated at different stages of the Vula migration project. This sub-subprocess is considered optional and can be bypassed by the respective management teams associated with the Units during the identification of individuals for migration.

5.4.2 (f) List of subprocesses for *Migrate scheduled users to an OSS desktop (A5)*

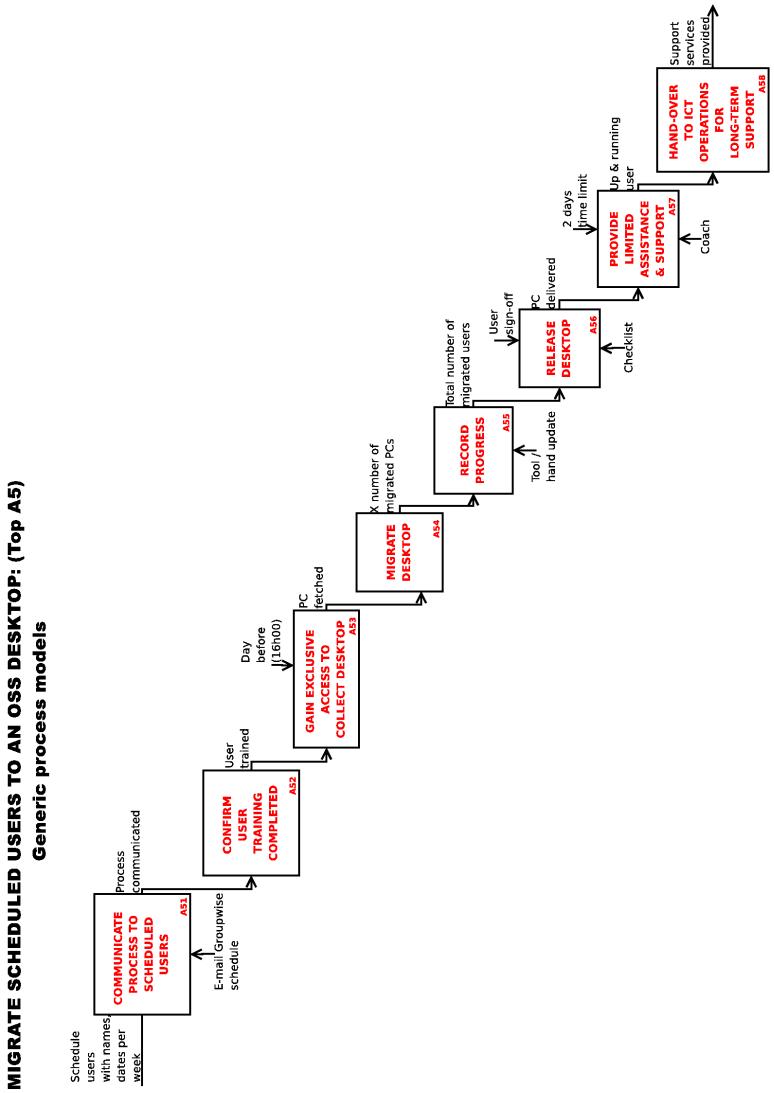

Figure 5.12: Highlighted process reference models for *Migrating scheduled users to an OSS desktop (A5)*

A51= Communicate process to scheduled users
A52= Confirm user training completed
A53= Gain exclusive access to collect desktop
A54= Migrate desktop

A55= Record progress
A56= Release desktop
A57= Provide limited assistance and support
A58= Hand over to ICT operations for long-term support.

Process Name	**Generic (Y/N/C)**	**Motivation**
Communicate process to scheduled users (A51)	Y	This activity is generic because it will help to eliminate any confusion and surprises which can potentially affect the success of the project. Thus, the process of the actual migration of the desktop as well as the expectations from the owner will be communicated to all the users.
Confirm user training completed (A52)	Y	This activity is generic because it confirms the completion of open source desktop training underwent by selected users scheduled for migration. Once training has been specified as completed, the user's desktop can be migrated. The initial goal of training is to enable selected users scheduled for migration to familiarise themselves with the new open source desktop and allow them to function productively on completion of the migration and also on the return of their desktop.
Gain exclusive access to collect desktop (A53)	Y	This activity is generic because once the desktops have been collected, they will enable or serve as a reminder to ICT technicians of the type of a job they need to do on those machines, if the desktops are not collected then this can potentially affect the success of the project.
Migrate desktop (A54)	Y	This activity is generic because it explains the actual process of how the desktops that have been brought in will be physically installed and started up, at the same time allowing the ICT technicians to perform various activities related to the actual migration on those desktops.
Record progress (A55)	Y	This activity is generic because once the above mentioned activity has been successfully completed, a database containing all the relevant information regarding the migration will be updated accordingly. The database will consist of the number of users that have migrated and those that have not done so with specific reasons given.
Release desktop (A56)	Y	This activity is generic because it has to do with the delivery of the desktop. The desktop can only be delivered to the office of the user after the completion of the 2-day open source desktop training.

Provide limited assistance and support (A57)	Y	This activity is generic because it explains the process of how users will be assisted after the migration by appointed, skilled coaches when experiencing any problems. This assistance is especially for those that will be struggling to utilise the new desktop and cannot apply the skills they learnt from training. Thereafter, the user will continue to use the desktop as per normal daily use.
Hand over to ICT operations for long-term support (A58)	Y	This activity is generic because this is where the ICT support line will continue to carry out their responsibility of supporting all the users within the organisation.

Table 5.12: Process reference models for migrating users

Forty users per week can be regarded as a quite large number for migration per week but that was the number chosen by the Vula project as its goal to meet. Other organisations may consider migrating a smaller number of users per week as their goal. Prior to the actual scheduled migration date, the process of the actual migration of the desktop as well as the expectations of the owner (or user of the desktop), were well communicated to all the users whose desktops were collected and prepared for migration. The intention was to eliminate any confusion and surprises which could potentially affect the success of the project. In addition to erasing any confusion, a desktop document drawn and agreed upon by the whole project team, describing the full functionality and shortcomings of the new OSS desktop was used give to the users. Therefore this subprocess *(A51) Communicate process to scheduled users* is considered generic because communicating the actual information to users deemed ready to migrate is of importance, in order to erase any concerns or confusion they may have.

Once that is done, it is essential to ensure that users undergo training before the desktops can be migrated. Once training is specified as completed, the user's desktop can be migrated. The goal is to ensure that scheduled users familiarise themselves with the new OSS desktop and allow them to function productively with the return of their desktop.

The next step involving the beginning of the actual migration will be to fetch the desktops of those users scheduled for migration. The collected desktops will enable or serve as a reminder to the ICT technicians (or ICT support staff responsible for the installations) of the type of job they need to do on those machines. Failure to collect desktops on time can potentially affect the success of the project and users' productivity.

After the collection, the desktop is brought to a pre-determined location where ICT technical staff will perform various activities related to the actual migration on it.

The total number of migrated desktops must be kept as a record on a database, to measure the progress or the increasing number of users being migrated on a weekly basis.

After the completion of the 2-day user training, the desktop is delivered to the office of the user as per scheduled appointment for delivery. Upon receival of the desktop, the machine will be physically installed and started up. The user can now proceed to apply all he/she learnt during the user training in his/her daily work. Once the desktop has been turned on the user must sign a document to show that the desktop was received in a good condition. During this time the users will relocate their data to their desired data structure.

After completion of the above, the user proceeds to use the desktop for normal daily use. It is expected that more assistance will be required initially for a maximum of 2 days, during which time migration coaches will be deployed in the respective units to assist immediately, should there be any problems experienced. The coaches is fully skilled to assist with all aspects of the Vula desktop.

Once all the above steps have been completed and the user is comfortable using the Vula desktop, the responsibility for support is then handed over to the ICT 2nd line Operations group (or ICT support staff) as part of their standard support portfolio. The ICT support staff will continue to do their normal jobs of supporting all the users within the organisation as they used to, but this time with additional knowledge of an OSS environment. The support services will be part of the maintenance plan discussed in the Support and maintenance process below to reveal how the support services will be provided to users once their desktops have been migrated to OSS.

(v) List of subprocesses for *Support and maintenance (A6)* process

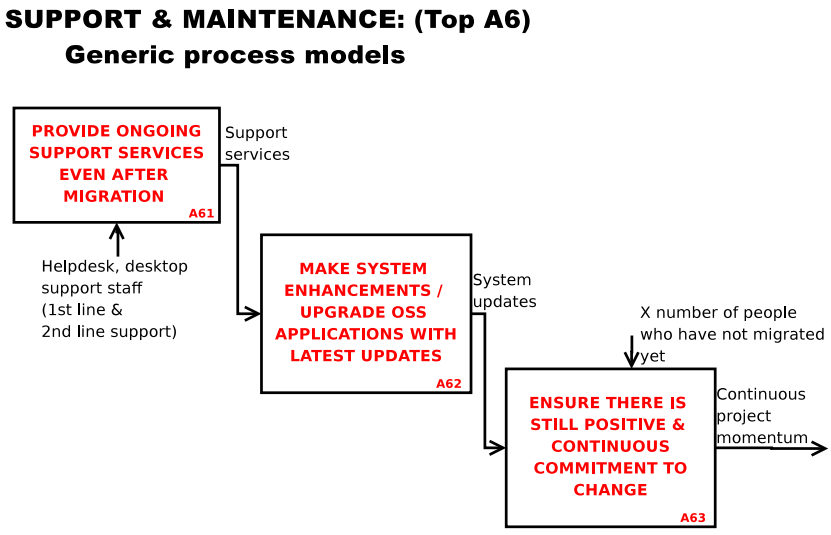

Figure 5.13: Highlighted process reference models for *Support and maintenance (A6)* process

A61= Provide ongoing support services even after migration
A62= Make system enhancements, upgrade OSS applications with latest updates
A63= Ensure there is still positive and continuous commitment to change.

Process Name	**Generic (Y/N/C)**	**Motivation**
Provide ongoing support services even after migration (A61)	Y	This activity is generic because even after the migration the current proprietary desktop must still be supported including the new open source desktop environment.
Make system enhancements, upgrade OSS applications with latest updates (A62)	Y	This activity is generic because it will ensure that all systems are updated with latest versions of new software releases.
Ensure there is still positive and continuous commitment to change (A63)	Y	This activity is generic because there will still be a large number of users left to migrate, so keep the momentum high and ensure that those that have already migrated are happy using the new desktop, as you would not want anyone to sabotage all the hard work you have done thus far.

Table 5.13: Process reference models for *Support and maintenance (A6)* process

Even after the migration the current Microsoft Windows desktop must still be supported with the new OSS desktop environment to ensure that

users' problems are resolved without much delays. That means the Helpdesk, ICT technicians or support staff must be fully equipped or skilled to perform the tasks of resolving issues related to the OSS environment. They form an important team that will need to run around and help users with their requests for help.

Systems enhancements is another factor that cannot be avoided during the support and maintenance process as it has to be done regularly to ensure that all the applications run as expected. That means all systems (e.g. servers, HR, ERP systems etc.) and its associated applications must be up to date with the latest versions of new software releases and running efficiently so as not to disrupt or affect the organisation's productivity in any way.

Bear in mind there will still be a large number of users left to be migrated. So it is advisable to keep the positive momentum high and ensure that those that have already migrated are happy in terms of using the new desktop. You would not want anyone to sabotage all the hard work and successes that have been achieved thus far, especially by those that have already migrated. Whatever mistakes are pointed out by the early adopters must be tackled as soon as possible to deliver good results and to avoid discouraging not yet migrated users.

5.4.2 (g) List of subprocesses for *Document lessons learnt (A7)*

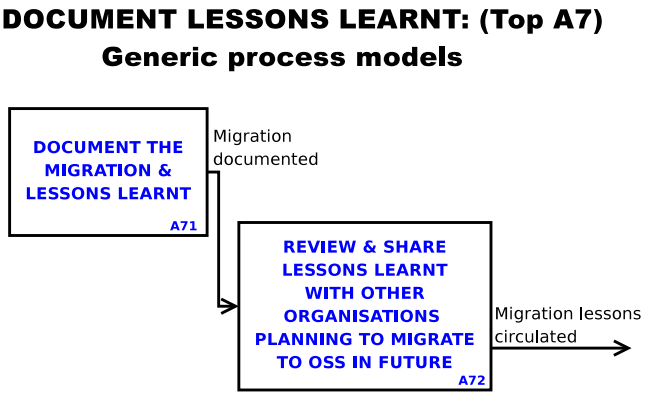

Figure 5.14: Highlighted process reference models for *Document lessons learnt (A7)*

A71= Document the migration and lessons learnt
A72= Review and share lessons learnt with other organisations planning to migrate to OSS in future.

Process Name	Generic (Y/N/C)	Motivation
Document the migration and lessons learnt (A71)	C	This activity is neither generic nor non-generic because not all companies will choose to document their projects. However documenting projects like open source migration can be very useful to organisations and businesses that might consider migrating to an open source environment in future. The documentation will guide them on how to go about the migration and help to avoid any risks involved.
Review and share lessons learnt with other organisations planning to migrate to OSS in future (A72)	C	This activity is also neither generic nor non-generic because it is a choice that can be made by the company. Sometimes sharing this information will create opportunities for companies to expand their relationships and collaborate or have talks with different types of organisations planning to migrate to a new environment or distribution in future.

Table 5.14: Process reference models for *Document lessons learnt (A7)*

The migration document can be a very useful document to the organisations that might consider migrating to an OSS environment in future. This documentation will guide them on how to go about the migration and to avoid any risks involved with the process. In return this document will create an opportunity for the CSIR to expand its relationships with other organisations, collaborate or have talks with different types of organisations planning to migrate to an OSS environment.

With so much experience gained during the migration of such a big organisation to OSS, the CSIR is set to act as a guiding example to other organisations. The CSIR can show them that this type of migration is doable. An organisation like the CSIR will feel very proud to provide a helping hand and the information that will help other organisations to migrate to OSS without any hassles. Even if other organisations are adopting a different distribution they could still make use of some of the data stated in this document. The documentation will basically provide anyone interested with project stages which were followed as part of the CSIR migration to OSS, especially those that were identified as generic.

5.5 VALIDATION

To analyse or verify the results of data established in the above Section 5.4, a focus group discussion was conducted with several Vula project experts to ensure that the extracted process reference models are a representation of what happened in the OSS migration project. In addressing these process reference models for an OSS desktop migration, it is possible to generalise the findings by saying the extracted process models are reusable or can be reused by organisations and companies embarking on a similar type of project. However, it

is important to test the findings of this research study in other environments, in order to establish that the findings of this research can be generic in other environments.

It is also possible that any other organisation might consider taking a different route from the one suggested here to plan and complete its OSS migration. It is however recommended that an organisation do its own research first and align the emerging results with the organisation's needs before beginning with the project. If the organisation lacks the time, resources and skills needed for this kind of a project, it is wise to consider other options such as relying on consulting organisations that have fine-tuned the migration process, the CSIR being one of them.

5.6 SUMMARY

This chapter provided a conclusion to the two research questions namely:

1. What are the key processes within an organisational open source migration?

2. What are the process reference models that are essential for an organisational open source migration?

These two research questions addresses the generic OSS migration process models (or process reference models for an OSS migration) which were the focus of this research. However, it was concluded that the process reference models for an OSS migration that were suggested in this chapter, can be tested for reusability against other case studies of the same nature to prove their validity.

Chapter 6

CONCLUSION

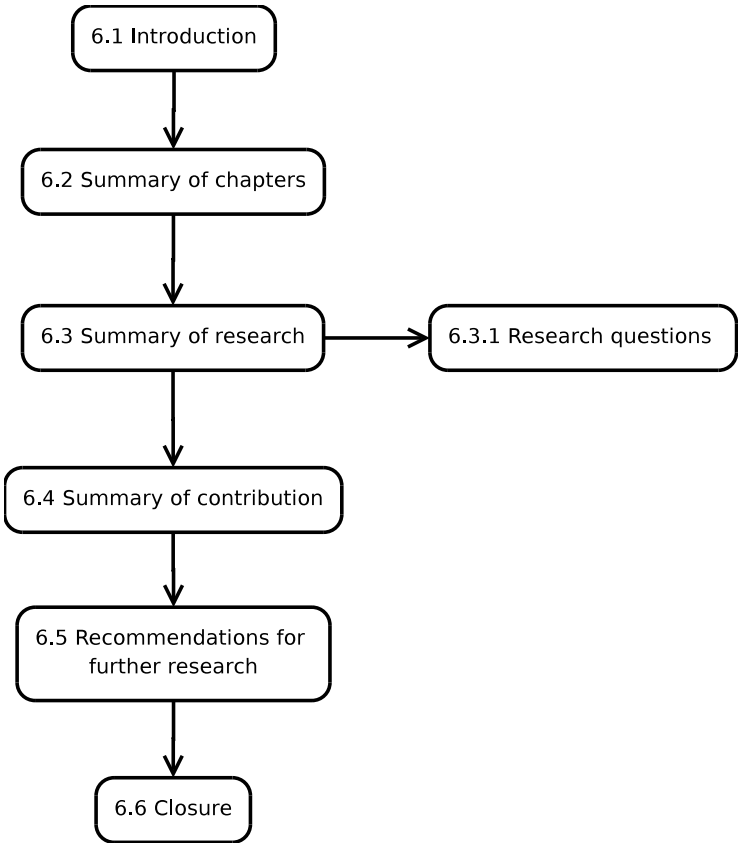

Figure 6.1: Chapter 6 outline

6.1 INTRODUCTION

The purpose of this chapter is to provide the conclusion for this study and to summarise the experiences and findings of this research study. This chapter is organised into four sections. The first section provides a summary of the chapters for the research study. The second section provides the results of this study and a summary for each of the research questions is given. The third section provides the summary of the contributions made by the study and the fourth section provides recommendations to organisational project leaders and teams considering a migration to the OSS environment. The fifth section provides further suggestions for future research and the dissertation concludes with the closure in the sixth section.

6.2 SUMMARY OF CHAPTERS

This study consists of six chapters and one annexure provided on a CD.
Chapter 1 provides introductory material to the study, highlights the background of the study and gives an overview of the problem and purpose for this research, namely, regarding the process reference models for an OSS migration.
Chapter 2 explores the theoretical overview of the usefulness of OSS and process reference models, as well as the role they both continue to play in an organisational context.
Chapter 3 provides background on the research design and methodologies used during the study, including an overview of qualitative research methods, that were used to collect the CSIR OSS migration data.
Chapter 4 focusses on establishing and discussing the process model structures for an OSS migration.
Chapter 5 focusses on the analysis of data articulated in Chapter 4, through the extraction of process reference models from a set of captured process models. These process models are identified from a case study (CSIR as the studied environment) through research participant interviews, questionnaires and literature reviews and are validated through a focus group discussion.
The study concludes with Chapter 6 and one annexure consisting of the questionnaire, interview data, documented migration processes data, as well as verified process reference models, which are all included on the CD.

6.3 SUMMARY OF RESEARCH

The purpose of this research study is to suggest a set of process reference models for an organisational OSS migration, which were identified using the systematic research approach by van der Merwe and Kotzé (2008). The process reference models were extracted from a set of process models, using the process modelling notation called IDEF0. This research was initiated to provide a reusable set of process models that can be used to plan and execute an OSS migration project. Many organisations and companies are migrating to OSS because of financial considerations. The process reference models identified in the research study intends to assist project managers and their teams to understand the process of migrating from proprietary software to OSS. The aim is that the process reference models be as generic as possible, once tested in other environments, so

that it can be easily adapted by not yet migrated organisations. The study was conducted at the CSIR within the South African context. Due to the shortage of skills and a competitive industry when it comes to OSS adoption in South Africa, the CSIR embarked on this project.

In order to ensure that open source technologies and software tools are adopted and given support in South Africa, organisations are urged to be inspired by the project of the CSIR migration to OSS and apply the lessons learnt onto similar projects in future. As stated in previous chapters, OSS is perceived to be better than proprietary software because of three important reasons: it is cheaper, customisable and secure.

The study is classified into four research questions addressed below:

- What is the use and value of a process reference model?
- What do we know about open source migration processes?
- What are the key processes within an organisational open source migration?
- What are the process reference models that are essential for an organisational open source migration?

The first research question, *What is the use and value of a process reference model?* was answered in Chapter 2 by the theoretical overview of a process reference model structure to obtain information on the usefulness of extracting generic process models from processes modelled with a process modelling tool.

In the second research question, *What do we know about open source migration processes?* several case studies of organisations that have migrated to OSS are reviewed. These case studies expose OSS migration projects which have already been performed worldwide. South Africa will not be the first nation to adopt OSS, but in previous projects the migration processes have not been documented.

The third research question, *What are the key processes within an organisational open source migration?* addresses the processes that took place during the OSS migration, using the CSIR as a case study for extracting such data. The migration processes varies from technological processes, communication processes, implementation processes to administrative ones. Research question 3 is answered by the identification and capturing of process models representing the processes of an organisational OSS migration.

The fourth research question, *What are the process reference models that are essential for an organisational open source migration?* reveals the possible generic migration processes (that is process reference models) which were extracted from a set of process models identified during the CSIR migration project to OSS. These process reference models are regarded as those that are bound to reoccur when it comes to OSS migration projects in other organisations.

6.4 SUMMARY OF CONTRIBUTIONS

This study makes two contributions. Firstly, it identifies and captures OSS migration process models. Secondly it suggests a set of process reference models extracted from those captured OSS migration process models. This set of process reference models is a generic process model structure that will assist project managers and teams involved in OSS migration projects to plan and execute this type of projects better in future.

These process reference models were validated using a focus group discussion, which was conducted with several Vula project experts to assess the validity of captured OSS migration process models. The objective is to ensure that the captured OSS migration process models is a representation of what happened in the CSIR's OSS migration project. These process reference models will in future be tested against other case studies of the same nature to prove their validity.

6.4.1 Lessons learnt

The main lessons learnt from this research study are:

- That it is important to encourage the adoption of OSS within organisations and government departments due to the cost reduction and security benefits associated with OSS (Cerri and Fuggetta, 2006; Dudley et al., 2006; GoOpenSourceTaskTeam, 2003; Hoe, 2007; Opensourceafrica.org, 2007; Simon, 2005; van Reijswoud and Topi, 2003).
- That it is important to capture OSS migration process reference models, as they will give the advantage of reducing mistakes associated with this kind of process if done the same way each time (Barn, 2007; SmartDraw.com, 2009; Tyrrell, 2000).
- That the shifting of users' mindsets should be considered when migrating from a proprietary to an OSS environment and it must be ensured that users are receptive to change Ahmed (2005); Dudley et al. (2006).
- That there should be support available from multiple vendors in terms of OSS, open standards and open formats to reduce the risk associated with vendor lock-in (Cerri and Fuggetta, 2006; Fuggetta, 2004; Shewale, 2009; Simon, 2005).
- That an OSS migration project can take time, hard-work and skills to complete (Ahmed, 2005; Belle et al., 2003).
- That you need buy-in from the decision makers, that is, top management (CEO, Executives and managers). They will play an important role, according to their leadership styles, in bringing about change in an organisation. That way they will have an effect on the success of a migration project (Ahmed, 2005).
- That providing training is essential for the success of a migration project (Ahmed, 2005; Horstmann, 2005; van Reijswoud and Topi, 2003).

- That it is imperative to capture the migration process for future reference in a way that can be easily adapted by not yet migrated organisations, without having to design their own from scratch (Rosa et al., 2005; van der Aalst et al., 2003).

6.5 RECOMMENDATIONS

As an outcome of this research, at least two recommendations can be made to emerging OSS migration project managers and teams.

The first recommendation is that the suggested process reference models listed in this research study must be tested in a similar kind of environment (at about 4 to 5 companies) before it can be adapted by organisations and companies migrating to OSS in future.

Another recommendation is that the questionnaire developed during the project can be used in a similar study of emerging OSS migration projects.

6.6 CLOSURE

Thus, this study supports the fact that a process reference models structure for an OSS migration can be established to reduce the uncertainty and fear in organisations planning to migrate from proprietary software to OSS, because there will be a document that can serve as a guide to help with the planning and implementing an OSS migration project.

Bibliography

AALBC.com: 2007, Glossary of publishing terms. Last accessed 15/04/2009.
URL: *http://aalbc.com/writers/publishing_glossary.htm*

Ahmed, O.: 2005, Migrating from proprietary to open source learning content management systems.

Allison, J.: 2009, The free software definition. Last accessed 19/11/2009.
URL: *http://www.gnu.org/philosophy/free-sw.html*

alnresearch.org: 2003, Structured interviews. Last accessed 15/04/2009.
URL: *http://www.alnresearch.org/HTML/AssessmentTutorial/Strategies/StructuredInterviews.html*

analytictech.com: 1997, Glossary of key terms. Last accessed 15/05/2009.
URL: *http://www.analytictech.com/mb313/glossary.htm*

Astor, A. and Rosenberg, D. K.: 2005, The top-five mistakes to avoid when migrating to open source, *Enterprise Open Source Journal* .

Barn, B.: 2007, Business process modeling.

Barnard, D.: 2007, South african government adopts open source policy. Last accessed 16/01/2009.
URL: *http://www.davidbarnard.blogspot.com*

Barry, D.: 2000, Business process execution language (bpel). Last accessed 24/8/2007.
URL: *http://www.service-architecture.com/web-services/articles/business_process_execution_language_for_web_services_bpel4ws.html*

Bechan, U.: 2008, Towards a framework for securing a business against electronic identity theft.

Belle, J.-P. V., Brink, D., Roos, L. and Weller, J.: 2003, Migrating to oss-on-the-desktop: Lessons learnt and a proposed model.

Berntsen, K. E., Sampson, J. and Osterlie, T.: 2004, Interpretive research methods in computer science.

Bider, I.: 2002, revised in 2003, Choosing approach to business process modeling - practical perspective. Last accessed 20/03/2009.

Borgatti, S.: 2001, Introduction to grounded theory. Last accessed 17/03/2009.
URL: *http://www.analytictech.com/mb870/introtoGT.htm*

Brain, D., Seltsikas, P. and Tailor, D.: 2005, Process modelling notations for egovernment: An assessment of modelling notations for identity management.

Browne, J. and O'Sullivan, D.: 1995, Re-engineering the enterprise. Last accessed 13/03/2009.
URL: *http://books.google.co.za/books*

Bulusu, G. K.: 2003, Ask the enterprise linux expert: What is the difference between open source and open standards? Last accessed 22/10/2009.
URL: *http://searchenterpriselinux.techtarget.com/expert/KnowledgebaseAnswer/0,289625,sid39_gci1052807,00.html*

BusinessDictionary.com: 2007, Unstructured interview definition. Last accessed 15/04/2009.
URL: *http://www.businessdictionary.com/definition/unstructured-interview.html*

BusinessRanks.com: 2006, Process modeling - process modeling resources. Last accessed 27/7/2007.
URL: *http://www.businessranks.com/process-modeling.htm*

BusinessTimesOnline: 2005, Open formats make history and maintain it. Last accessed 3/11/2009.
URL: *http://business.timesonline.co.uk/tol/business/industry_sectors/technology/article579744.ece*

Centre_for_Ecological_Sciences: 2009, Welcome to free and open source software (foss) at ces - faqs.

Cerri, D. and Fuggetta, A.: 2006, Open standards, open formats, and open source, *The Journal of Systems and Software* p. 80.

Childe, S. J., Smart, P. A. and Weaver, A. M.: 1997, The use of generic process models for process transformation, pp. 1 – 10.

Collins-Brown, E.: 2006, Aspects of online courses that are more effective and successful than traditional, face-to-face courses, pp. 77 – 81.

CSU, C.: 1993, Glossary of key terms. Last accessed 17/03/2009.
URL: *http://writing.colostate.edu/guides/research/glossary/*

Dalziel, J.: 2003, Open standards versus open source in e-learning, *EDUCAUSE QUARTERLY* (4), 4 – 7.

de Paula, V.: 2006, Beyond the classroom lecture series notes - business modeling with uml, **2**(2), 1 – 9.

Dibben, K.: 2004, Online communities educause. Last accessed 21/11/2008.
URL: *http://209.85.173.104/search?=cache:oyvhvzof6mkJ:www.educationau.edu.au/jahia/webdav/site/myjahiasite/shared/papers*

Dortch, M.: 2009, Capturing and documenting business processes. Last accessed 20/11/2008.
URL: *http://www.ebizq.net/topics/bpm/features/8086.html?rss*

dst.gov.za: 2006, Open source software freedom day celebrations at the department of science and technologys (dst). Last accessed 02/10/2009.
URL: *http://www.dst.gov.za/media-room/press-releases/open-source-software-freedom-day-celebrations-at-the-department-of-science-and-technology2019s-dst-new-headquarters/*

Dudley, G., Finlay, A. and Otter, A.: 2006, Open source software: The five step migration guide. Last accessed 25/02/2008.
URL: *http://www.opensourceafrica.org/default.php?view=migration guide*

Economic and (ESDS), S. D. S.: 2007, Esds qualidata teaching resource: exploring diverse interview types. Last accessed 15/04/2009.
URL: *http://www.esds.ac.uk/qualidata/support/interviews/unstructured.asp*

Elvis, M. and Mathur, S.: 2008, Literature reviews. Last accessed 17/04/2009.
URL: *http://explorations.sva.psu.edu/lapland/LitRev/litrev.htm*

epractice.eu: 2007, Government backs open formats by egoveditorialteam. Last accessed 3/11/2009.
URL: *http://www.epractice.eu/en/news/281947*

Eriksson, H.-E. and Penker, M.: 2000, *Business Modeling with UML - Business Patterns at Work*, John Wiley & Sons, Inc.

fao.org: 1990, Semi-structured interviews. Last accessed 15/04/2009.
URL: *http://www.fao.org/docrep/x5307e/x5307e08.htm*

F_D_Bluford_Library: 2005, How to do a literature review? Last accessed 17/04/2009.
URL: *http://www.library.ncat.edu/ref/guides/literaturereview03.htm*

Fettke, P., Loos, P. and Zwicker, J.: 2005, Business process reference models: Proceedings of the workshop on business process reference models (bprm 2005).

FIPS_PUBS: 1993, Announcing the standard for integration definition for function modeling (idef0), *Draft Federal Information Processing Standards Publication 183* .

Fitzgerald, B. and Kenny, T.: 2003, Open source software in the trenches: Lessons from a large-scale oss implementation.

Fogwill, T., Bezuidenhout, H., Verheij, R. and Basson, E.: 2007, Notes on the current status of the csir migration to open source and open standards through project vula, pp. 1 – 5.

Fogwill, T., Bezuidenhout, H., Verheij, R. and Basson, E.: 2008, Csir open source migration project status update.

Foster, D.: 2008, Maintaining a successful corporate community. Last accessed 23/10/2008.
URL: *http://fastwonderblog.com/2008/09/17/maintaining-a-successful-corporate-community/*

Fremantle, P.: 2003, Open source versus open standards. Last accessed 22/10/2009.
URL: *http://pzf.fremantle.org/2008/05/open-source-versus-open-standards.html*

fsfe.org: 2009, What is free software? Last accessed 19/11/2009.
URL: *http://www.fsfe.org/about/basics/freesoftware.en.html*

Fuggetta, A.: 2004, Open source and free software: A new model for the software development process?, *The European Journal for the Informatics Professional* **5 / 2004**(5), 227 – 238.

Genzuk, M.: 2003, A synthesis of ethnographic research.

go opensource.org: 2005, Open source software. Last accessed 17/04/2007.
URL: *http://www.go-opensource.org/software/*

GoOpenSourceTaskTeam: 2003, Sa national open source strategy - the south african view on open source.

Halpin, T.: 2006, Entity relationship modeling from an orm perspective: Part 1, *Journal of Conceptual Modeling* . This is a revised version of a paper that first appeared in the December 1999 issue of the Journal of Conceptual Modeling, published by InConcept.

Hammersley, M.: 2008, Observation, participant and non-participant. Last accessed 15/05/2009.
URL: *http://www.blackwellreference.com/public/tocnode?id=g9781405124331_chunk_g978140512433121_ss1-2*

Havey, M.: 2005, *Essential Business Process Modeling*, O'Reilly Media, Inc.

Herman, E. and Bentley, M.: 1993, Rapid assessment procedures (rap): Conduct unstructured interviews. Last accessed 15/04/2009.
URL: *http://www.unu.edu/Unupress/food2/UIN04E/uin04e00.htmContents*

Hoe, N. S.: 2007, Free and open source software - origins, benefits, myths and realities, *The Malaysian Open Source Group* .

Holetzky, S.: 2003, What is a case study? Last accessed 06/03/2009.
URL: *http://www.wisegeek.com/what-is-a-case-study.htm*

Holland, B.: 1996, Guide s47 - process modelling guidelines. Last accessed 17/04/2007.
URL: *http://www.for.gov.bc.ca/his/datadmin/s47.htm*

Horstmann, J.: 2005, Migration to open source databases, *Computation and Information Structures (CIS)* pp. 12–13.

IBM: 2006, Integrating open source into your business.

in eLearning UCEL, U. C.: 2004, Glossary of key terms. Last accessed 15/05/2009.
URL: *http://www.ucel.ac.uk/rlos/p_np_observation/2.html*

Jensen, C. and Scacchi, W.: 2003, Reference model based open source software process discovery.

JISC-InfoNet: 2007, Process review. Last accessed 09/02/2009.
URL: *http://www.jiscinfonet.ac.uk/InfoKits/process-review/process-review-6*

JRS-Consulting: 2007, A glossary of market research terms. Last accessed 26/10/2009.
URL: *http://www.jrsconsulting.net/freearticles_21.html*

KBSI: 2006, Idef0 overview by knowledge based systems inc. (kbsi). Last accessed 25/02/2008.
URL: *http://www.idef.com/idef0.html*

Kings_College_London: 2009, Qualitative research glossary.

Kok, F.: 2005, Open source software as a value alternative to commercial software.

Kovács, S. D. . G. L. and Kochis, P. Z.: 2005, Risk assessment of an open source migration project.

Krechmer, K.: 2005, Open standards requirements.

Kulandaisamy, P. D.: 2004, Model driven legacy migration, pp. 1–6.

Leonardo, Z.: 2004, Critical social theory and transformative knowledge: The functions of criticism in quality education, **33**(6), 11–18.

Malhotra, Y.: 2006, Business process redesign: An overview. Last accessed 10/02/2009.
URL: *http://www.brint.com/papers/bpr.htm*

Martinez, A.: 2005, Open source, a development option.

Michael, F.: 2009, Process modeling purpose. Last accessed 17/03/2009.
URL: *http://www.bpmnforum.net/blog/bpmn/bpmn-in-practice/process-modeling-purpose/*

Mokhema, T.: 2005, Go open source task team conference results of 22 and 23 august 2005 - the go open source campaign partners, government departments, the private and public sectors and academia. Last accessed 16/01/2009.
URL: *http://www.meraka.org.za/news/goopensourceconference_8sep 2005.html*

Moore, A., McQuay, H., Derry, S. and Moore, M.: 2007, Qualitative and quantitative research: Bandolier glossary. Last accessed 06/03/2009.
URL: *http://www.medicine.ox.ac.uk/bandolier/booth/glossary/qualres.html*

Munro, R., Negrette, J. and Aitken, A.: 2003, Engaging with the open source community (part one). Last accessed 18/11/2008.
URL: *http://www.theinquirer.net/en/inquirer/news/2003/06/22/engaging-with-the-open-source-community-part-one*

Myers, M. D.: 1999, Investigating information systems with ethnographic research, **2**(23).

Myers, M. D.: 2004, Qualitative research in information systems. Last accessed 09/01/2008.
URL: *http://www.qual.auckland.ac.nz/*

NADbank.com: 2008, Glossary of key terms. Last accessed 15/05/2009.
URL: *http://www.nadbank.com/en/resource/glossary*

NCREL: 2008, Ethnographic research. Last accessed 09/03/2009.
URL: *http://www.ncrel.org/sdrs/areas/issues/educatrs/profdevl/pd5lk2.htm*

Neill, J.: 2007, Qualitative versus quantitative research:key points in a classic debate. Last accessed 06/03/2009.
URL: *http://wilderdom.com/research/QualitativeVersusQuantitativeResearch.html*

NETC.org: 2003, Open options. Last accessed 22/10/2009.
URL: *http://www.netc.org/openoptions/background/word.html*

NMSU, N. M. S. U.: 2008, Glossary of key terms. Last accessed 15/05/2009.
URL: *http://planning.nmsu.edu/taskforce/glossary.html*

Novell-Versora-Linux-Team: 2005, Moving to a linux desktop: A novell and versora migration study, pp. 1–40.

open std.org: 2002, Open standards repository. Last accessed 3/11/2009.
URL: *http://openstandards.org/*

opendocumentfellowship.com: 2009, Opendocument in government. Last accessed 3/11/2009.
URL: *http://opendocumentfellowship.com/government*

Opensourceafrica.org: 2007, Open source software - the five step migration guide.
URL: *http://www.opensourceafrica.org/default.php?view=migrationguide*

opensource.org: 2009, About the open source initiative (osi). Last accessed 26/11/2009.
URL: *http://www.opensource.org/about*

oss.gov.za: 2006, Policy on free and open source software use for south african government. Last accessed 17/04/2007.

Pawlak, M., Bryce, C. and Lauriére, S.: 2004, Environment for the development and distribution of open source software: A project reference model for edos, **2**(5.5.3), 1–168.

Peeling, N. and Satchell, J.: 2001, Analysis of the impact of open source software, pp. 4 – 5.

Perens, B.: 2007, Open standards principles and practice. Last accessed 3/11/2009.
URL: *http://perens.com/OpenStandards/Definition.html*

Perumpalath, B.: 2005, Modelling business process: An integrated approach.

Pesic, M. and van der Aalst, W.: 2005, Towards a reference model for work distribution in workflow management systems, pp. 1–82.
URL: *http://is.tm.tue.nl/staff/wvdaalst/publications/p278.pdf*

Public_Health_Agency_of_Canada: 2005, Glossary. Last accessed 17/04/2009.
URL: *http://www.phac-aspc.gc.ca/canada/regions/atlantic/Publications/Making_connections/4-eng.php*

RedHat_Inc.: 2007, Open source adoption. Last accessed 16/11/2007.
URL: *http://www.redhat.com/truthhappens/public_policy/osa/*

Reference.com: 2006, Encyclopedia topics. Retrieved on 10/02/2009 from.
URL: *http://www.reference.com/browse/wiki/Artifact*

Rosa, M. L., Gottschalk, F., Dumas, M. and van der Aalst, W. M.: 2005, Linking domain models and process models for reference model configuration. Last accessed 25/02/2008.

Rosemann, M., Recker, J., Green, P. and Indulska, M.: 2007, Evaluation and acceptance of process modelling grammars. Last accessed 17/10/2007.
URL: *http://www.bpm.fit.qut.edu.au/projects/acceptance/*

Schreiber, A. and Wielinga, B.: 2005, Knowledge model construction. Last accessed 12/03/2009.
URL: *http://ksi.cpsc.ucalgary.ca/KAW/KAW98/schreiber/*

Sharma, A.: 2003, Tracking the open source front. Last accessed 17/04/2007.
URL: *http://www.technetra.com/writings/archive/2003/07/02/tracking-the-open-source-front*

Shewale, N.: 2009, Importance of open access, open source, and open standards for libraries, pp. 1–101.

Silva, D. L.: 2006, Lecture 9 process modelling part 1.

Simon, K. D.: 2005, Integrating open source into your business, *IBM SYSTEMS JOURNAL* **44 / 2005**(2), 227 – 238.

sita.co.za: 2006, Free open source software (foss) - strategies and plans. Last accessed 05/10/2009.
URL: *http://www.sita.co.za/FOSS/FOSS1.html*

SmartDraw.com: 2009, Document the existing processes - tutorials. Last accessed 23/10/2009.
URL: *http://www.smartdraw.com/resources/tutorials/Document-the-Processes*

Smuts, J.: 2008, Towards a comprehensive knowledge management system architecture.

sociology.org: 2007, Structured interviews, pp. 1–4.

Sotirios Koussouris, Aikaterini Maria Sourouni, J. P. K. K.: 2007, Generic process models for e-business transactions in heterogeneous systems.

Soy, S. K.: 1997, The case study as a research method. Last accessed 26/03/2009.
URL: *http://www.gslis.utexas.edu/ ssoy/usesusers/l391d1b.htm*

SparxSystems: 2007, Uml tutorial - the business process model. Last accessed 28/7/2007.

Stanford-Medical-Informatics: 1996, Supply-chain operations reference-model (scor). Last accessed 25/02/2008.
URL: *http://www.supply-chain.org/cs/root/scor_tools_resources/ scor_model/scor_model*

Sun, Y.: 2006, An introduction to process modeling. Last accessed 13/02/2009.
URL: *http://yang-sun.com/modeling.aspx*

Sutcliffe, A.: 2002, The domain theory: Patterns for knowledge and software reuse. Last accessed 13/03/2009.
URL: *http://books.google.co.za/books*

Taraborelli, D.: 2004, Why use open formats? Last accessed 3/11/2009.
URL: *http://www.openformats.org/main*

Tellis, W.: 1997, Introduction to case study. Last accessed 26/10/2009.
URL: *http://www.nova.edu/ssss/QR/QR3-2/tellis1.html*

Texas_Department_of_Information_Resources: 2008, An introduction: Open standard, open source, open format. Last accessed 18/11/2009.

Thomas, D.: 2005, Going open source software in it opportunities and challenges, *Journal of Object Technology* **4**(2), 7–13.

Turbit, N.: 2005, Business process modelling overview. Last accessed 17/7/2007.
URL: *http://www.projectperfect.com.au/info_business_process_model ling_overview.php*

Tyrrell, S.: 2000, The many dimensions of the software process.

ubuntu.com: 2004, About the ubuntu organization. Last accessed 28/01/2009.
URL: *www.ubuntu.com*

UNESCO: 2007, Open source and proprietary software.

University_of_Texas: 2007, Instructional assessment resources (iar) - glossary. Last accessed 26/10/2009.
URL: *http://www.utexas.edu/academic/diia/assessment/iar/glossary.php*

USEPA: 2007, Program evaluation glossary. Last accessed 27/03/2009.
URL: *http://www.epa.gov/evaluate/glossary/r-esd.htm*

US_Homeland_Security: 2008, Process reference model for assurance mapping to cmmi, pp. 1 – 9.

Uvium_Inc.: 2005, Business process management and optimization glossary. Last accessed 03/02/2009.
URL: *http://www.uvium.com/business-process-management-glossary.cfm*

van der Aalst, W., Dreiling, A., Gottschalk, F., Rosemann, M. and Jansen-Vullers, M.: 2005, Configurable process models as a basis for reference modeling - business process reference models position paper, pp. 75–81.

van der Aalst, W. M., Dreiling, A., Rosemann, M. and Jansen-Vullers, M.: 2003, Configurable process models as a basis for reference modeling. Last accessed 26/01/2009.

van der Merwe, A. J.: 2005, Towards a reusable process model structure for higher education institutions. Last accessed 26/01/2009.

van der Merwe, A. and Kotzé, P.: 2008, A systematic approach for the identification of process reference models.

van der Merwe, A., Kotzé, P. and Gerber, A.: 2007, Generic process model structures: Towards a standard notation for abstract representations, pp. 172–180.

van Reijswoud, V. and Topi, C.: 2003, Alternative routes in the digital world - open source software in africa.

VulaWiki: 2006a, Project vula. Last accessed 17/04/2007.
URL: *https://vula.csir.co.za*

VulaWiki: 2006b, Vula progress report 1. Last accessed 19/11/2007.
URL: *https://vula.csir.co.za/index.php/Project_VulaVula_progress_report*

Weiss, D.: 2005, Measuring success of open source projects using web search engines.

West, J.: 2004, What are open standards? implications for adoption, competition and policy.

Yin, R. K.: 2003, *Case Study Research: design and methods*, Sage Publications, 3rd edn. London, England.

Appendix A: Picture of CSIR Documents

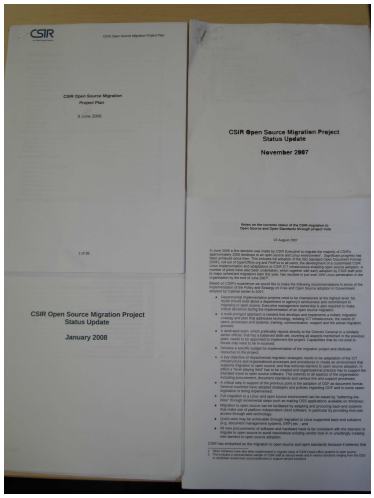

Figure 6.2: Examples of the CSIR documentation for the OSS migration project

Appendix B: List of publications and links on Open Source

It is important to note that the links listed above were last visited on April 24, 2009.

Organisation Name	Links
Women'sNet organisation moves to OSS. It was founded in 1997 with the aim of empowering Southern African women to use ICTs.	www.womensnet.org.za
Open Source Software Migration at the Orwell High School in UK	http://www.osor.eu/case_studies/open-source-software-migration-at-the-orwell-high
Migration to OSS at Beaumont Hospital Dublin in Ireland	http://www.osor.eu/case_studies/docs/beaumont-hospital-dublin
Open Source Software Migration in the Belgian City of Schoten	http://www.osor.eu/case_studies/open-source-software-migration-in-the-belgian-city

Other case studies can be found at:	• http://www.tectonic.co.za/view.php?id=1877 • http://www.tectonic.co.za/view.php?id=1407 • http://www.opensourceafrica.org/default.php?view=case_studies • http://www.li.org/success/ • http://www.opensource.org/advocacy/case_studies.php • http://www.itpapers.com/cgi/SubcatIT.pl?scid=262&wc=3 • http://www.redhat.com/solutions/info/casestudies/ • http://www-3.ibm.com/software/success/cssdb.nsf/topstoriesFM?

Table 6.1: URLs or websites of already migrated organisations

Appendix C: Questionnaire

This questionnaire consists of six parts as pointed out in Section 3.3.2.4 and it is provided on the CD with all the questions which were included in the questionnaire.

Appendix D: Interview Questions and Responses

Provided on CD.

Appendix E: High-level and refined levels of the CSIR migration to OSS

The focus of Appendix E is on documented migration process models for the OSS migration project. Due to space limitations only a few of the graphic representations of the processes, subprocesses and atomic processes (or sub-subprocesses) were presented, the rest of the data gathered can be viewed on the accompanying CD. These processes and subprocesses inputs, outputs and goals were highlighted in the form of a table. In addition, a brief description of each of the processes and subprocesses were provided after the graphical representation of the relevant process.