JYVÄSKYLÄ STUDIES IN COMPUTING 21

Anja Mursu

Information Systems Development in Developing Countries

Risk Management and Sustainability Analysis in Nigerian Software Companies

Esitetään Jyväskylän yliopiston informaatioteknologian tiedekunnan suostumuksella julkisesti tarkastettavaksi yliopiston vanhassa juhlasalissa (S212) kesäkuun 27. päivänä 2002 kello 12.

Academic dissertation to be publicly discussed, by permission of the Faculty of Information Technology of the University of Jyväskylä, in Auditorium S212, on June 27, 2002 at 12 o'clock noon.

UNIVERSITY OF JYVÄSKYLÄ

JYVÄSKYLÄ 2002

Information Systems Development in Developing Countries

Risk Management and Sustainability Analysis in Nigerian Software Companies

JYVÄSKYLÄ STUDIES IN COMPUTING 21

Anja Mursu

Information Systems Development in Developing Countries

Risk Management and Sustainability Analysis in Nigerian Software Companies

UNIVERSITY OF JYVÄSKYLÄ

JYVÄSKYLÄ 2002

Editors
Seppo Puuronen
Department of Computer Science and Information Systems, University of Jyväskylä
Marja-Leena Tynkkynen and Pekka Olsbo
Publishing Unit, University Library of Jyväskylä

URN:ISBN:9513912914
ISBN 951-39-1291-4 (PDF)

ISBN 951-39-1210-8 (nid.)
ISSN 1456-5390

ABSTRACT

Anja Mursu
Information systems development in developing countries: risk management and sustainability analysis in Nigerian software companies
Jyväskylä: University of Jyväskylä, 2002, 296 p.
(Jyväskylä Studies in Computing,
ISSN 1452-5390; 21)
ISBN 951-39-1291-4
Finnish summary
Diss.

The expectations of IT for socio-economic and human development in developing countries have been high, but the real benefits and the positive impacts have been somewhat disappointing. The harsh socio-economic and infrastructural context makes the sustainability of an IS a major issue. Technology transfer to Africa has been too dependent on external factors, like international aid donors, and emphasized on technology itself. There is a need for locally developed, appropriate information systems, which are based on local needs and structures. In addition, the development should be done by using appropriate methods in an African context. This thesis is part of the INDEHELA-Methods project, where we have studied ISD in Nigerian software companies. This has been done by a survey, by case studies in software and user companies, and by a risk study of software projects. This thesis reports the case studies and the risk study in detail. The survey is introduced shortly. The cases are analysed by using an activity theoretical approach as a framework. The risk study repeats an international study conducted earlier in the US, Finland and Hong Kong, by using the Delphi method. The main differences in software development when compared to the industrial countries are in a socio-economic and infrastructural context, rather than in technical issues. The software business in Nigeria is professional and ambitious, and capable of providing IT solutions for local companies, but the resources for IT investments are scarce. The study provides a model for risk management and sustainability analysis for project management.

Keywords: information system, information systems development, risk factor, risk management, sustainability, activity analysis, developing countries, Africa, Nigeria

ACM Computing Review Categories

H.1.0 Information systems: Models and principles: *General*

H.1.1 Models and principles: Systems and information theory: *General systems theory*

K.2. History of computing: *Hardware, people, software, systems, theory*

K.4.1. Computing milieux: Computers and society: Public policy issues: *Computer-related health issues, ethics*

K.4.3 Computing milieux: Computers and society: Organizational impacts: *Automation, employment, reengineering*

K.6.1 Management of computing and information systems: Project and people management: *Systems analysis and design, systems development*

K.6.3. Management of computing and information systems: Software management: *Software development, software process*

Author's address Anja Mursu
University of Jyväskylä
Department of Computer Science and Information Systems
P.O.Box 35, FIN- 40014 University of Jyväskylä, Finland
University of Kuopio
Department of Computer Science and Applied Mathematics
P.O.Box 1627, FIN- 70211, Kuopio, Finland
e-mail: anjamu@it.jyu.fi, anja.mursu@uku.fi

Supervisors Prof. Kalle Lyytinen
Department of Information Systems
Case Western Reserve University, USA

Docent Mikko Korpela
Computer Centre, HIS R&D Unit
University of Kuopio

Reviewers Prof. Robert Davison
Department of Information Systems
City University of Hong Kong, Hong Kong

Prof. Markku Nurminen
Computer Science
University of Turku, Finland

Opponent Dr. Richard Heeks
Institute for Development Policy and Management
University of Manchester, UK

ACKNOWLEDGEMENTS

The starting point of my research ‘chapter of life’ was when Ms. Maritta Pirhonen proposed to me to work with her as an instructor on the Development Project Course at the University of Jyväskylä, Department of Computer Science and Information Systems. As Seppo Kortelainen, the CEO of the late Relatech Ltd. and my boss at that time, kindly conceded a leave of absence for me, I started to work for the university in 1996. I also had the pleasure to work with Mr. Eero Tourunen and professor Eero Peltola at the department.

In 1997 I met Dr. Mikko Korpela, who was looking for doctoral students for the prospective project concerning informatics in Africa. At that time my interest was in software project risk management, but I found Korpela’s project extremely interesting. We noticed that we could easily combine our interests. Professor Kalle Lyytinen accepted my new research context without question, and was ready to provide his supervision. So I started as a full-time researcher on the INDEHELA-Methods project in August 1998. The project was headed by docent Dr. Korpela from the University of Kuopio, Computing Center, HIS (Health Information System) R&D Unit.

The Academy of Finland is acknowledged for funding the project for four years altogether from 1998-2001. The Computing Centre provided a half year position to me to complete this manuscript and COMAS Graduate School has given me some grants for traveling expenses. I am grateful for their financial support.

I would like to thank my two supervisors, professor Kalle Lyytinen and docent Mikko Korpela. Professor Lyytinen’s advices, comments and relevant questions have been directed to take care of the rigourous angle of my work. Dr. Korpela’s participation in the empirical parts has been invaluable. Of course their roles have not been that distinct, but their support has been most important in every way.

It would have been impossible to conduct this research without the cooperation of my Nigerian counterparts from the Obafemi Awolowo University (OAU) in Ile-Ife. Especially I would like to thank Ms. Abimbola Soriyan, who is completing her thesis within the project at the time of writing this text. We have spent many interesting, exciting and memorable moments in the field, in work and in pleasure. Ms. Soriyan has been an essential part when collecting empirical data in Nigeria. I also like to thank professor Akinde for his support and encouragement, and Ms. Karen Olufokunbi for her contribution in the early phases. There are also a number of students at the OAU, who have helped with data collection in the risk study. I appreciate their assistance, and I especially want to thank Mr. Adewale Adeqaye and Mr. A. Afolabi.

My special thanks go to the companies that have participated in this research, anonymously. I am very grateful especially to the case companies and their client companies for the interesting and enthusiastic interviews. It was not possible to circulate the case descriptions for review in all the companies, so I apologize for any possible misconceptions.

The INDEHELA-Methods project has worked in near collaboration with other researchers under the ENHICA (environment and health in West Africa) research collaboration, for example by publishing a joint book. I would especially like to thank professor Marja Järvelä (University of Jyväskylä, Dept. of Social Sciences and Philosophy), and her doctoral students Ms. Kristiina Kuvaja, Ms. Lea Hausen and Ms. Eva-Marita Rinne for very pleasant collaboration and discussions. Ms. Kuvaja offered accommodation for my very first night in Nigeria, and Ms. Hausen and Ms. Rinne as pleasant company during my first visit to Nigeria, not forgetting Mr. Matti Koskivirta. With Eva we have managed to fix some of our later visits to Nigeria in unison, thus having an opportunity to share experiences and enjoy the peaceful afternoon moments together.

There are many colleagues who have been interested and also curious about my research, thus encouraging the work. In the following I would just like to mention a few. Ms. Anne Honkaranta, Ms. Riitta Kuisma and Ms. Nazmun Nahar have proved to be excellent company in the conference journeys, in sharing research experiences, but also in the late night 'improving the world' discussions. Ms. Eija Karsten has acted as a senior researcher, but more as a friend by giving her support and good advice concerning the writing process. Ms. Maritta Pirhonen has been my colleague and a friend for years, and I would like to thank Maritta and Kari for their support. Professor Seppo Puuronen was a great help when preparing this manuscript for printing, I sincerely appreciate his comments for improving the manuscript. I also thank professor Pekka Neittaanmäki for his help in the printing process.

I am very thankful to professor Robert Davison and professor Markku Nurminen for their work as external reviewers of the thesis. Dr. Richard Heeks has promised to be an opponent in the public defence. In the way of my research I have received valuable comments and critics from Lars Mathiassen, Kristin Braa, Ojelanki Ngwenyama, Tom McMaster, Brian Fitzgerald, among others. Also many anonymous and peer reviewers have given high-quality comments for articles this thesis is based on. Roy Schmidt provided practical help in the risk study. Gary Littler helped with improving the language.

Finally I would like to thank my family and all my friends for their encouragements, support, and interest concerning my work, and I thank Anssi for everything, not least for coming to Nigeria with me for a month.

Eija adviced me in Boise in summer 2001, that writing a thesis is 'once in a lifetime matter, enjoy it!'. So I did.

Jyväskylä
May 2002

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

"When you come to a waterfall
Mist blowing against your face and arms
Do you stop and ask for understanding at these moments
I said not yet but I intend to start today"
Raymond Carver (1989); *A New Path to the Waterfall*, p. 113

This study traverses through the fields of *information systems research*, *development studies*, and *activity theory*. In information systems research, my original interest was in software project risks, and furthermore the failures of information systems. I was lucky to be able to combine my research interests in an uncommon context, by studying these features in a developing country, thus bringing the development aspect into the picture. My work forms part of a research project (INDEHELA-Methods) where a whole service chain of systems development is being examined in a developing country in West-Africa, Nigeria; from the university of education, through software engineering and systems development to the use of information technology. At the start of the work I was thinking whether my interests would be appropriate for the research topics of the project. The appropriateness seemed to be adequate since the project interest was in systems development in developing countries, not only in technology transfer, or software import. In addition, what would be a more problematic place for systems development than a developing country like Nigeria. During the early phase of my work I noticed that how could I understand the risks or failures in software development in Nigeria if I did not know how the systems are actually developed there. How do I know how the risks are affecting the systems development work? Since there are very few studies about information systems development in Africa, I decided to extend my research to systems development work in software companies. I also noticed that besides managing risks in software development, equally important is how to consider risks in the use of these systems. New technologies are easily abandoned for example in Africa if they are not appropriate. Here we use the term sustainability of an information system to denote a successful and viable introduction of information technology in a user organization. The third field of my study is the activity theory. Activity theory is used as a background theory for analysing systems development as a work activity. Thus these three fields are combined by conducting a *study of information systems development in Nigeria by applying the activity-theoretical approach*.

Why study information systems development in developing countries? Somebody may ask whether it is even appropriate to study such things for example in Sub-Saharan Africa when there are more 'obvious' problems like famine, diseases like AIDS or malaria, over population, poverty, and so on. The question of the relevance of informatics studies has not been relevant for a long time for development researchers, since information technology (IT) has promoted (developed) socio-economic well being in many sectors, like industry, education, trade, health care, agriculture, and traffic, especially in the industrialised countries. Why should we in the industrialized countries have the privilege to make use of modern technology for our own well being, while people in developing countries do not? Only because they are poor? It is essential in this discussion to make a difference between development and modernization, even if they both refer to a societal change process (see Järvelä & Kuvaja 2001). Modernization represents a change which can be both positive or negative; it does not evaluate the nature of societal change. Modernization exists in every society, but its nature varies from one society to another. The concept of development, in turn, presents a positive social and economic change. In order to achieve development by modern IT, the new computerised system should be sustainable and appropriate to its use environmentally.

The key message of the World Development Report 1998/99 for Africa (World Bank 1999a) is that most countries on the continent need to do much more, and much faster, in order to increase their knowledge base, to invest in educating people, and to take advantage of the new technologies for acquiring and disseminating knowledge. The important role of information technology for Africa's development has recently been recognized by several international agencies, for example:

- Within the United Nations the African Information Society Initiative (AISI) is the African mandate to use information and communication technology to accelerate economic and social development in Africa (http://www.bellanet.org/partners/aisi/).
- The Information for Development Program (infoDev) is a global program managed by the World Bank to help developing economies fully benefit from modern information systems. (http://www.worldbank.org/infodev/infodev.html)
- *USAID* has dedicated itself to empowering developing countries with the knowledge and technology to not only survive but to thrive in the electronic global climate. (http://www.info.usaid.gov/info_technology/)
- In the International Development Research Centre, Canada, the Acacia Initiative is an international effort to empower sub-Saharan African communities with the ability to apply information and communication technologies to their own social and economic development. (http://www.idrc.ca/acacia/)

There is another question, which I believe, is relevant here. Why should people in developing countries develop information systems by themselves while there are plenty of systems provided by industrialized countries? Or “does it make sense for African nations to set up their own software industry” (Heeks 1996a, p.5)? As I have mentioned already, the role of IT for socio-economic development and growth in developing countries has been found to be critical, and software forms a vital part of this (Odedra-Straub 1996, Waema 1996, Heeks 1996a). Still, the desired development of nations has been fulfilled inadequately, since the IT applications have not been successfully applied for example in Africa (Odedra-Straub 1996). For example, according to Odedra-Straub, IT has not been applied to sectors which influence the majority of the population in developing countries. Development theorists, politicians, and even information system (IS) specialists have often expressed their uncertainty about the appropriateness of IT for achieving development goals (see Avgerou & Land 1992). In that sense, the question of introducing IT is not about the ‘transfer of technology’ alone, but adopting and developing suitable software technology and applications locally, to suit particular local needs (see e.g. Heeks 1996a). Also, the technology transfer to Africa has been too dependent on external factors, and emphasized on technology itself. That is why countries in Africa – like Nigeria – need to adapt and develop their own ways of using information systems based on their own needs and processes, using their own methods and practices. The “appropriate development is an endogenous increase of the capabilities of a community to decide on its future and to implement its decisions” (by Pellegrini 1980 in Korpela 1995 p. 21). I also agree that “all major IT projects must have local content and involvement for sustainability” (a notion made in the Software Exhibition in Lagos 2001). Information system development (ISD) thus forms an essential part of the diffusion and implementation of IT in developing countries.

We can still ask two questions: is systems development work so different in developing countries when compared to industrialised countries that it is worth studying? And also, are there systems development activities in Sub-Saharan Africa in the first place? To these questions I am going to provide answers in this work. I will also continue to explore the first question by asking what are the differences and what causes these differences. Is it because of the culture, political environment, material facilities or what? Are the problems different in Nigeria when compared to industrialised countries? Avgerou (1996) suggests that ISD practices and methodologies are not universal, but need to be adjusted to any given socio-economic, cultural, and organisational setting. Do the companies in Nigeria have such adjusted methodologies, and if not, should they have one?

In addition, I agree with Korpela (1994) that information technology in Africa forms a mirror that lets us look at the ethics, constraints, benefits, and drawbacks of information technology from a global point of view.

In the following sections I will first introduce the INDEHELA-Methods project in terms of research objectives and goals. After that I present my own research problems, objectives, and questions. The research design and methodology of the project is illustrated in the next section. I will put forward my own research methods in more detail in the methodology chapter of this thesis. The theoretical background of the thesis is clarified next, in terms of research interest, research context, and research approach, with the references to the structure of this thesis. The intended contributions of the study are presented next, including theoretical contribution, knowledge contribution, practical contribution, and methodological contribution. In the summary I introduce the overall view of the study. The structure of the thesis and the list of publication of the project as related work complete this chapter.

1.1 INDEHELA-Methods project as a background

This work is part of the INDEHELA–Methods project (Methods for Informatics Development for Health in Africa) by the Department of Computer Science and Engineering of the Obafemi Awolowo University (OAU), Nigeria, and the HIS R & D Unit, Computing Centre of the University of Kuopio, Finland. The project was established to develop a 'Made-in-Nigeria' ISD methodology which addresses the special needs of Nigerian systems developers. Healthcare is used as the main field of application in practical experiments and examples. The project was funded by the Academy of Finland between 1998-2001.

The INDEHELA-Methods project is based on the Ife Project (Korpela et al. 1998), which is a joint project of health informatics research and development in Nigeria, between the two above named universities, and with the OAU Teaching Hospitals Complex, since the 1980s. Rudimentary Hospital Information System, MINPHIS (Made-in-Nigeria Primary Healthcare and Hospital Information System), was developed in 1989, and the system has been in operation in one of the hospital units of the OAU Teaching Hospital since 1991 and another hospital unit since 1995. The system has been further developed during the years, and the cooperation and system development still continues. In this practical project the goal is to develop a local hospital information system that will be diffused to other hospitals in Nigeria as well in other African countries.

The core research group of the INDEHELA-Methods consisted of three people. The first one was a part-time senior researcher from the University of Kuopio, who was also overall project leader. There was one full-time researcher, the author of this thesis, working as a PhD student in Finland. In addition there was a part-time researcher and PhD student in Nigeria, who was also a local project leader. In Nigeria, another part-time researcher (MPhil student) and a couple of final year BSc students have participated in the methodological research at various stages. The local project leader in Nigeria is also coordinating the practical systems development project, MINPHIS. Also a number of other staff and final year students have participated in the MINPHIS project.

FIGURE 1 The central computer in the OAU teaching hospital

FIGURE 2 The computer centre of OAU

The *main objectives* of the INDEHELA-Methods project are, firstly, to produce empirical evidence and understanding of the practice and problems of ISD in Nigeria, and secondly, to facilitate the Nigerian ISD practitioners by improved methods, techniques, practices, and education. The study covers the whole service chain of systems development, as presented in FIGURE 3. The core research framework covers information systems development including software development (see definitions later in this thesis), the use of information systems in customer organization and how it affects the services they render to the society, thus having an impact on the society. At the other end of the chain is the information systems (IS) education at universities.

Education equips students with skills and knowledge, while IS research is required for instance to enhance education. The coloured circles in the FIGURE 3 illustrate the parts I am contributing within the whole framework.

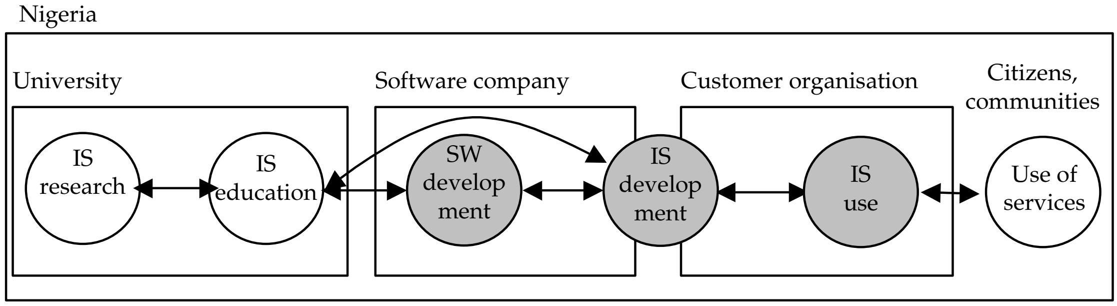


FIGURE 3 The core research framework of INDEHELA-Methods project (coloured parts represent my research interests)

The expected outcome of this thesis is part of the proposed improved ISD methodology by the INDEHELA-Methods project, which will be documented in Lecture Notes and tested in practice in MINPHIS project. Lecture Notes are meant to introduce and elaborate in systems analysis courses and final year projects of Computer Science students at the Obafemi Awolowo University.

Next I will present my contribution to the research framework in more detail.

1.2 Research problems, objectives, and questions

In this section I introduce my research problems, objectives, and research questions that directed my work. The intended results are also defined. The starting point of my research, the initial *research problem*, was a question whether the risks in software development are different in Sub-Saharan Africa when compared to industrialized countries. All the previous risk studies have been conducted in industrialised countries, and the risk management methods were based on the results from industrialised countries. Thus there was no knowledge about the appropriateness of existing risk management methods for African countries. At the same time we noticed that there is no knowledge about the systems development practices in Africa. We did not know whether ISD in Sub-Saharan Africa in general and in Nigeria in particular is different from ISD in the industrialised countries? If the answer is yes, what are the differences and what causes these differences? In order to understand the results of the risk study, we also had to find out how the risks are affecting systems development work in Nigeria? Accordingly, the *purpose of my study* was to examine what are the methods and practices applied in Nigerian software companies, and what are the risk factors involved in systems development. So far, most IS research in Africa concerns the constraints of IT introduction, and the failures to gain socio-economic development, or even operational effectiveness in user organizations. The importance of IT for development, and the emphasis of the difficulties in IS implementation and use confirmed that when developing ISs, equal emphasis should be placed into the use of ISs. We wanted to know, how the successful use of ISs can be already considered in the systems development process. We chose the concept of sustainability of IS to illustrate the successful use of IS. Thus we embedded a risk management and a sustainability analysis to be critical parts of a successful ISD process. Accordingly, the other purpose of my study was to examine, what the factors are that impact on the sustainability of ISs in use.

Thus the *overall objective* of the study is to indicate, how risks in information systems development projects can be identified, monitored, and mitigated and how sustainability of an information system can be facilitated by improved ISD methods (FIGURE 4). The *overall research problem* is, whether there is a need for a new comprehensive perspective for sustainable development of ISs?

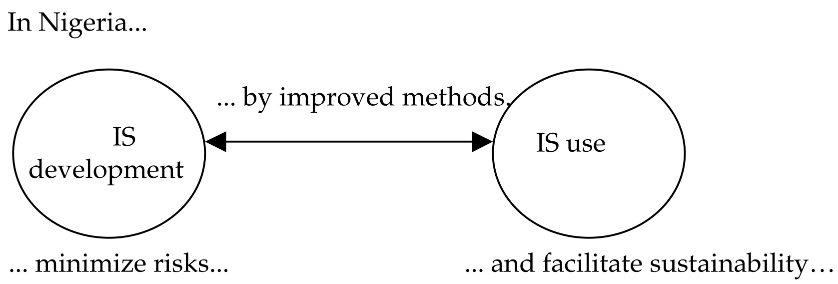


FIGURE 4 Overall research objective

In order to achieve the overall research objective, we need to formulate more detailed research questions concerning organizations as well as software business in general (FIGURE 5):

- **Information systems development**: How many companies have ISD activities in Nigeria? What kind of resources do the companies have? What kind of methods, techniques, and practices are used? How is the work organized?
- **Software development**: What kind of risks are involved in software development? What kind of methods, techniques, and practices are used for development work?
- **Information systems in use:** What kind of companies and other organizations are using ISs? What kind of benefits do the companies or organizations reach from ISs for themselves, for their personnel (users), and for the clients? How much new ISs changed work practices? How was this change organized? What kind of resources were needed for implementing new ISs? What kind of resources are needed for maintaining ISs? What are the factors effecting the sustainable use of ISs?

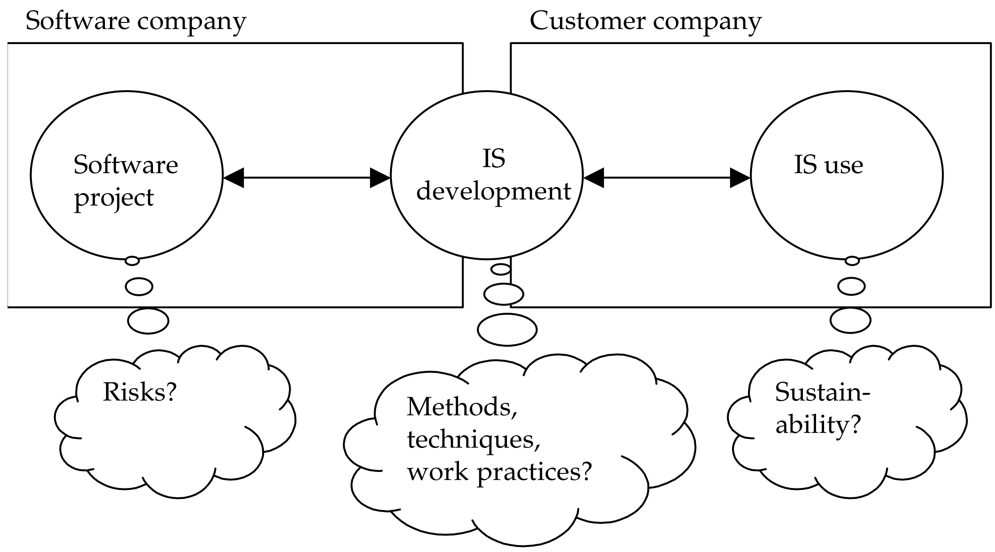


FIGURE 5 Detailed research questions

The results of the study should provide empirical evidence and thus answers to the above mentioned research problems and research questions. In addition, the results should provide empirical data which can be used, together with a literature review, as a basis for the improved method for ISD. The research objective presented in FIGURE 4, method for risk management and sustainability analysis, is supposed to be realized in Lecture Notes, where the adapted methodology, provided by INDEHELA-Methods project, for information systems development is presented. Thus the implications of the study are for practitioners with Lecture Notes, and for other researchers and also practitioners with obtained knowledge of risk factors and ISD practices in Nigeria, as well as the sustainability factors.

1.3 Research design and methodology

The research methodology of my work forms part of an overall research design of INDEHELA-Methods project. The overall research design is to develop Made-in-Nigeria systems development methodology as presented in FIGURE 6. The research has three inputs: 1. literature review of relevant ISD methodologies in general; 2. theoretical analysis of the special requirements in Africa and Nigeria; 3. survey and case studies about existing practice and problems among Nigerian software companies. The inputs are used for formulating adapted ISD methods, techniques, and practices specifically for the Nigerian context, which are tested and further developed doing action research in one or two 'host projects'. The coloured boxes in FIGURE 6 indicate the parts of the overall research design that are processed in this thesis.

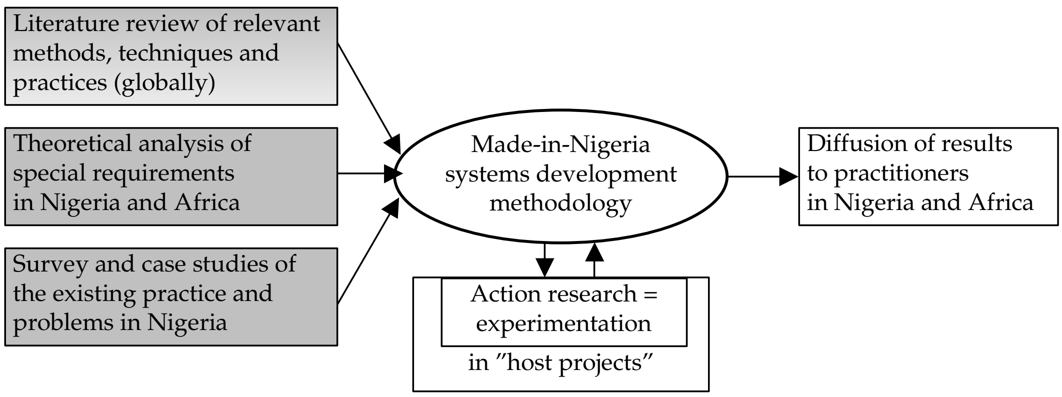


FIGURE 6 The overall research design of INDEHELA-Methods project (Korpela et al. 2000a, p. 144)

Since the objective of the project is to develop an applied methodology, relevant ISD methods, techniques, and practices developed in other countries must be identified through a literature review (Korpela et al. 2000b). Concerning the literature review, this thesis focuses more on the empirical studies on ISD practice. This kind of literature must be studied to acquire guidance concerning the research approach and methodology, as well as to clarify contributions. This way one can reflect the methodology she finally uses for others in order to see the hidden assumptions and possible weaknesses of the chosen approach. Also, the results can be better understood, since this kind of review increases the knowledge and insights of the issue. Besides the ISD studies, concerning the research questions, relevant methods of risk management and sustainability analysis will be reviewed.

Theoretical analysis must be applied when identifying areas where the requirements for and constraints of ISD in Nigeria and in Africa in general may differ significantly from those of the industrialised countries (Korpela et al. 2000b, Mursu et al. 2000). The development of the adjusted ISD methods and techniques can then focus on these areas.

An overall view of the existing practice and problems in ISD in Nigeria can be achieved through an interview survey among Nigerian software companies. The interview survey is divided into two parts: a questionnaire survey to obtain an overview of software industry in Nigeria and a risk study to collect the most common risk factors in software development in Nigeria. In this thesis, the questionnaire survey is presented mainly by referring to the work of the Nigerian colleague, Mrs. Soriyan, but the risk study is presented in detail. The overall view must be supplemented by descriptive in-depth case studies in a few companies and projects.

The three inputs will be merged together to an adapted methodology, a portfolio of methods and techniques, as a result of the INDEHELA-Methods project. Initially the methodology will be just a ‘prototype’ which must be tested and further developed through action research (Avison et al., 1999) in one or two real life ‘host projects’. The most accessible testing ground for the emerging methodology will be the MINPHIS project of OAU and OAUTHC (see INDEHELA-Methods project above). The methodology will also be tested in Systems Analysis courses and final year projects of Computer Science students at the university. Later on the methodology should be tested in a software company by systems practitioners, and further developed again according to the feedback. The actual testing and further development will be included for future research. The risk management and sustainability analysis method resulted from this thesis is part of the methodology.

Based on the overall research design presented in FIGURE 6, the research methodology to complete this thesis consists of two empirical studies as well as a literature review of the significant areas and theoretical considerations. The empirical studies concerning my research interests include a Delphi study to collect the most common and important risk factors of software development in Nigeria. The risk study repeats a design of the risk study conducted earlier in the US, Finland, and Hong Kong (Keil et al. 1998, Cule et al. 2000, Schmidt et al. 2001). We also asked IT experts to name factors for sustainable use of ISs within the first round of the Delphi study. In addition, we conducted more in-depth case studies to obtain knowledge of information systems development work in Nigeria, and to relate the software business to a societal context. Within the cases I have also included two interviews with the customers of software companies to attain knowledge of the systems development process from a client's perspective. These interviews provide additional interesting details, but they are not analyzed as case studies. Research methodology and process are described in more detail in Chapter 5.

1.4 Theoretical background

The research design and also the structure of this thesis is based on three main 'pillars':

1. IS and ISD research and practices, as *a research interest*.
2. Activity theory, as *a research approach*.
3. Development countries, more accurately Sub-Saharan Africa, as *a research context*.

1.4.1 Research interest

The research interest is based on IS and ISD traditions of research and practice, and to be more specific, on European research tradition or rather, on the Scandinavian research tradition, Scandinavian School. Information system research is a multi-discipline topic and as much a social, as a technical science. Its foundations can be found in philosophy, and in the organisational and behavioural sciences, as well as in mathematics and the natural sciences (Galliers 1994). It is important to make a difference between information technology (IT) and the information system (IS). Information technology refers to a software product, or an application package running in a computer based system. *Information system* refers to the information processing activities within an organization, which may or may not be facilitated by information technology. IS is a social system, having IT as one aspect (Land 1994, Walsham et al. 1990). *IS is IT in use*. Thus what makes IT 'appropriate' is the collective human activity in an organization within which it is utilized.

A respective distinction must be made between software engineering (SE) and information systems development (ISD). SE deals with the technical construction of the software product, while *ISD* deals with how to serve organizational purposes by means of software products. We consider information systems development as much a work development process in user organization, as a systems development.

It can be assumed that SE is more similar across countries, while ISD varies more according to the socio-economic and political settings (Korpela et al. 2000b, Mursu et al. 2000). There has been very little empirical research done on SE and ISD in developing countries. There is not much empirical evidence on whether information systems development is very similar or widely different in industrialized and developing countries.

The number of failures in ISD projects has been and is still high in spite of advances in technology and obtained experience (see e.g. Keil et al. 1998, Schmidt et al. 2001). In the IS literature, failures and the importance of *risk* management has been reported since the 1970s (see e.g. Alter & Ginzberg 1978, Lyytinen & Hirschheim 1987, Charette 1989, Boehm 1991, Cule et al. 2000). Usually failures in ISD refer to cost overruns, project delays, and unmet user needs (Barki et al. 1993, Cule et al. 2000). The first stage in managing failures is to identify risk factors. Several articles have been published concerning various types of risks associated with software projects (e.g. Boehm 1989, McFarlan 1981, Barki et al. 1993, Schmidt et al. 2001). By risk factor, we mean a contingency that can form a serious threat to the successful completion of an information systems development project (Barki et al. 1993). There is some literature about constraints in utilizing IT in developing countries (e.g. Waema 1996), but there is no comprehensive study of software development project risk in a developing country. Software production capabilities are building up in many African nations, and there are considerable software industries (Heeks 1999b). Thus the knowledge of the most common risk factors in ISD development projects in Africa would be valuable for any practitioner or researcher.

The failures of the ISD process relate closely the use of information system or information technology; the system is late or the requirements are not met and so on. Even if the development process itself is efficient and correctly implemented, the eventual use of IS can fail. The system is not used effectively, or it can even be abandoned for several reasons. The IS failure can even cause a serious accident like the space shuttle Challenger (e.g. Perrow 1984). In this work we are more concerned with systems that are hardly used by the users, since they are not convinced of their value (c.f. Lyytinen & Hirschheim 1987). In developing countries IT is expected to promote socio-economic or human development in the country, which expectation is not usually met (e.g. Odedra-Straub 1996). The challenges of the use of IT for development for example in Africa are so high, that it would not be reasonable to study the failures of ISD, and leave the use of ISs out of the study. Developing ISs is not an easy task, and understanding the failures in 'use' situation is important. We have chosen a term *sustainability* of ISs to illustrate the situation where the systems are appropriately used and are not creating contradictions in the user organization. In the 1990s there have been major global debates on sustainable development in all sectors of development. Sustainability of ISs is one part of this concept. In the case of IS and IT, sustainability means the ability to identify and manage risk factors threatening the long-term viability of the information system, or ISD activity. In development studies concerning ISs and IT, I argue that sustainability is one major concern.

In Chapter 2 I briefly explore the field of ISD. It includes an exploration of IS research and practice historically, aiming to clarify the role of the Scandinavian tradition in the field. The literature review of the related research focuses on the following studies:

- empirical studies of ISD practices, mainly looking for the used research methodologies and contributions of studies,
- empirical studies of risk factors and risk management approaches, and
- studies of failures in systems introduction, mainly aiming to identify factors for sustainable solutions.

1.4.2 Research approach

The research approach – activity theory – is 'a heritage' from the INDEHELA-Methods project. The project has applied a research framework named *Activity Analysis and Development* (ActAD), which is based on the activity theory (Korpela 1994, Korpela 1999). Here I present the main ideas and concepts of the activity theory as well as the ActAD framework. During the research process, the framework has been tested and further developed for the purpose of IS research.

In this study I apply the framework of ActAD as a research method for studying ISD as a work activity. Work has not been commonly used as a unit of analysis in information systems research (there are exceptions like Eriksson & Nurminen 1991). For example the concept of information, or the use of information in organizations has been more common. According to Kuutti (1994), one possible reason for that is the orientation of dominant background theories in ISD research that have not been *primarily* interested in work, e.g. phenomenology, hermeneutics, Habermas or Giddens. Systems development and the use of information systems are both very much work activities that include all the elements of the activity theory. For example, an information systems development project is a collective phenomenon, having a subject (usually collective subject) who transforms an object to an outcome – to some work which needs to be improved. The collective subject needs tools, or a means to carry out the work, like physical equipment and the development methodology. The work takes place in some community, which shares the object. There are rules controlling the work and communication, and a division of labour, project organization, roles, and so on.

Firstly in Chapter 3 I explain the advantages of using the activity theoretical approach for studying ISD compared to some other possible theories. After that I introduce the basics of the activity theory and the background and elements of Activity Analysis and Development. I also indicate what is different in ActAD when compared to an 'orthodox' model by Engeström (1987a), or other activity theoretical approaches. In the literature review I explore how the activity theory has been applied in information systems research. In Chapter 5 I apply the framework of ActAd when presenting the cases of my research.

1.4.3 Research context

The research context is also 'a heritage' from the INDEHELA-Methods project. The research takes place in Nigeria, with the cooperation between the University of Kuopio and Obafemi Awolowo University in Ile-Ife. The cooperation is based on the long-term Ife Project since the 1980s (Korpela et al. 1998), thus it is feasible to conduct this research study in Nigeria.

Nigeria is the most populous country in Sub-Saharan Africa with rich resources. It is a developing country (DC) in terms of GNP, which was 260$ per capita in 2000 (in US dollars, World Bank 2002). Still, with a population of around 127 millions people in 2000 (World Bank 2002), the domestic market for indigenous entrepreneurship, for example, in computer applications is large. In 1994, there were more than 500 registered computer companies offering computer-related services in Nigeria (Alabi 1994). We estimate that the figure should be more than 1000 today, of which maybe 20% are providing software development services (Soriyan et al. 2002). The standard and quality of computer education in Nigeria is high, especially in the field of software. The problems are very much the same when compared with other developing countries: poor infrastructure, weak economy, and an unstable society. The software industry or systems development has not been studied much in Sub-Saharan Africa. Thus we do not know what kind of practices or methods the companies are applying, or what kind of risks or contradictions there are for software development. Thus it is also reasonable to conduct this study in Nigeria. We do not suggest that Nigeria is a representative of a Sub-Saharan African country in terms of culture, but more in terms of environmental, infrastructural, and socio-economical context.

In Chapter 4 I define what is meant by developing countries and I make a short presentation of Africa and Nigeria, because the history and context of the continent is relevant in order to understand the results and contributions. The literature review consists of studies and articles of information technology in developing countries: what kind of issues have been emphasised and what are the conclusions and recommendations, especially in Sub-Saharan Africa. Nigeria is approached from the perspective of research problems. The survey conducted within the INDEHELA-Methods project is also presented in this chapter in order to get an overview of the software industry in the country.

1.5 Contributions of the study

As in many research efforts, we can list several contributions concerning the study: *theoretical contribution*, *knowledge contribution*, *practical contribution*, and *methodological contribution*. The contributions of this work are related to the fields I have described: information systems research, development studies, and activity theory. The main contribution focuses on information systems research, being mainly knowledge contribution, but also a theoretical contribution. The motivation of my study can be found from the practical contribution and development studies. Methodological contribution is a 'by-product', although not less important, at least for other researchers. In the following I will clarify these contributions of my study.

Theoretical contribution deals with information systems research by introducing a multidisciplinary study combining multiple theories of software risks and information systems failures, information systems development, information systems development in developing countries (developmental informatics), and the activity-theoretical approach. The theoretical analysis results in some requirements we, within the INDEHELA-Methods project, recommend to be taken into consideration in the ISD process and methodologies. The main theoretical contribution of my study extends software development risk management to also cover sustainability factors in the user organization. This study is a qualitative study trying to develop a shared understanding of a specific phenomenon within a specific cultural and contextual environment. Theoretical contribution is directed to IS academics and researchers both in industrialized countries and developing countries.

The results of the study provide the knowledge contribution of this study. The knowledge is related to information systems development:

- Knowledge of risk factors in software development that are considered important by IT-experts in Nigeria.
- Knowledge of ISD methods and practices in Nigerian software companies.
- Knowledge of factors that users consider important for successful implementation of a new information system.
- Knowledge of needed resources to maintain new information system in user organization.

Knowledge contribution is meant for both researchers and practitioners. Researchers can use the knowledge when they want to compare or reflect their own studies to other studies. For practitioners I wish to provide useful knowledge so that they can reflect on the results of their own work and hopefully find inspiration to improve their work activities.

Practical contribution is one of the 'driving forces' of this work. Practitioners in Nigeria are given useful knowledge (see above) about their own companies as well as software industry in the country. In addition, by documenting the results of the whole project into Lecture Notes, we are hoping to provide useful methods, techniques, and principles for software companies in order to help them improve their practices. In addition, Lecture Notes can serve as a means of education at universities by providing the same methods, techniques, and principles to be introduced in IS courses and by providing better knowledge of the current situation in the field and their problems. My work provides tools for risk management and sustainability analysis included in the Lecture Notes.

Methodological contribution is related to activity theory and an application of using the activity-theoretical approach to information systems research. The research work tests and further develops a method called Activity Analysis and Development (ActAD) to be suitable for work development researches in the field of information systems as well as in other disciplines. The framework can be used by researchers as a research methodology, or practitioners when analysing work practices, their own or others when they are working as facilitators or consultants. Thus ActAD can serve as a research methodology as well as a work development methodology. Besides testing the suitability of ActAD method for ISD research, the study tests the suitability of the Delphi method for conducting research in a context of a developing country.

1.6 Summary

The overall framework of this study is illustrated in FIGURE 7. The figure summarises the issues introduced in this chapter: theoretical background of the study, research objectives, and research methodology, and also the contributions of the study.

The middle part of FIGURE 7 illustrates research topics; software project risks, information systems development methods and practices, and sustainability factors in information systems use. The research takes place in Nigeria. Thus in order to understand the research object there must be some contextual investigation of organizations, culture, society, economy, and environment. This kind of examination must be conducted both empirically and by the literature review.

The upper part of the picture illustrates the theoretical background; research interest, research context, and research approach. The theoretical part strongly influences the research methodology, which is presented at the bottom of the picture. The Delphi method is used for the risk study, by questionnaire we can obtain an overview of the systems development in Nigeria, and interviews and ActAD method are used for creating an insight into a typical systems development project in a Nigerian software company. We also include customers of software companies (also users of information systems) in this analysis in order to achieve knowledge of the implementation and use of ISs.

Each level provides a specific contribution. Contributions are illustrated on the right part of the picture.

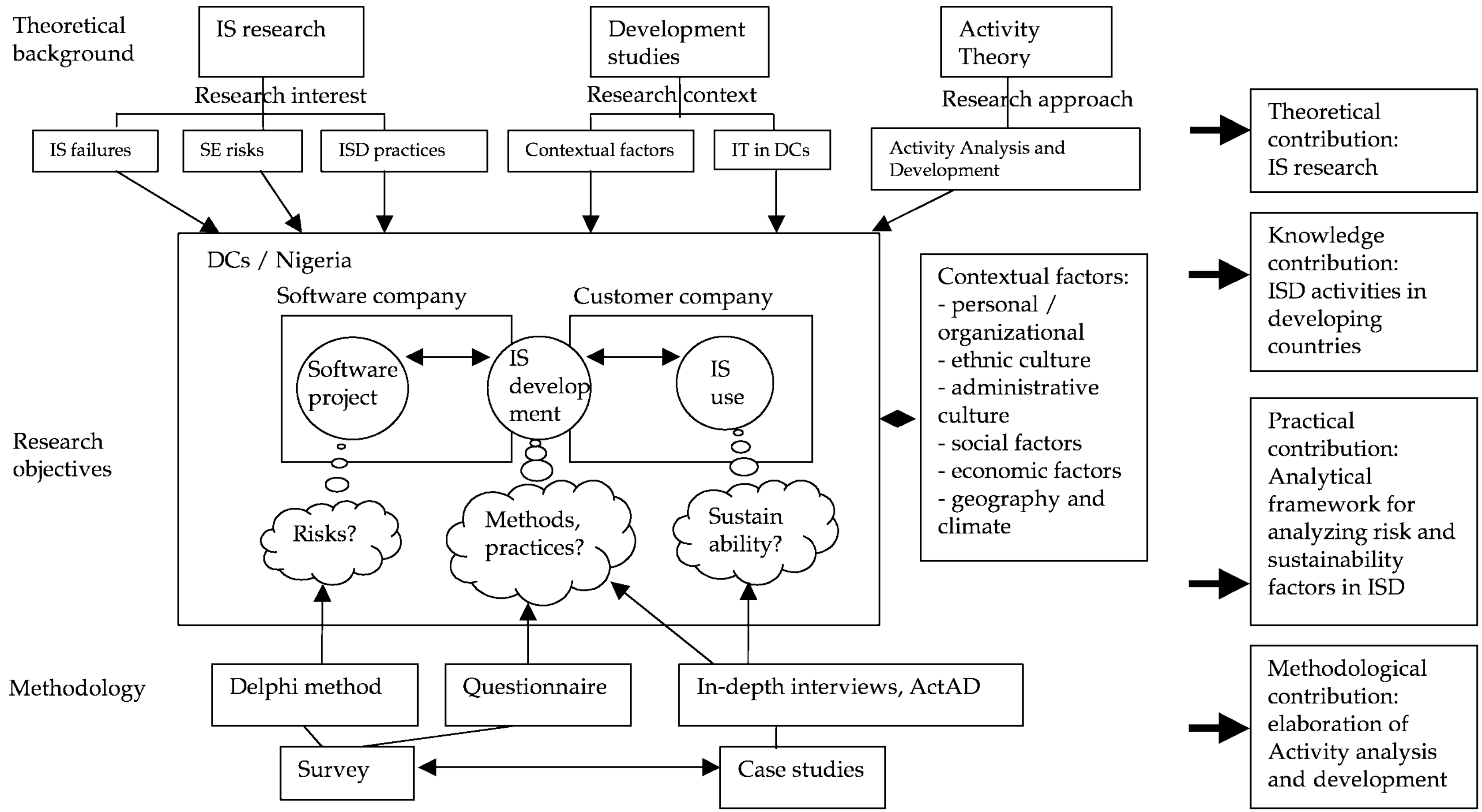


FIGURE 7 The overall view of the study

1.7 Structure of the thesis

In this chapter we present the general background of the study: the motivation, objective, research questions, research design, theoretical background, and expected contributions. The chapter seeks to explain and argue why these specific research topics are included in the thesis. Why is it relevant and reasonable to study information systems development in Nigeria? Why do we think it is important to study risks of software development and sustainability factors in IS use? How are the results going to be utilized in theory and in practice?

The next four chapters relate to the background of the study. Chapter 2 introduces the area of research interest. The chapter begins with a historical review of information systems research and information systems practice. The aim of the review is to clarify the role and position of Scandinavian studies to the IS research discipline in general. The Scandinavian studies are argued to be innovative and diversified in research theories and methodologies, they emphasize IS evolution and user-participation, apply mainly anti-positivistic approaches and are practical oriented. These arguments motivate us to be part of the Scandinavian tradition. After that empirical studies of information systems development (ISD) are examined as a related research. The main interest is in indicating used research methods in ISD studies, as well as reached contributions to the field. In addition, I have summarised what has been said about the failure or success of the ISD projects. Next I move to the risks of software development projects, and risk management methodologies. From software development risks I continue to consider risks in the use of ISs. The last issue in chapter 2 discusses the sustainability of ISs, and what it means in practice.

The next chapter, Chapter 3, introduces the basics of the activity theory and Activity Analysis and Development, as a research framework. Chapter 4 presents the research context in more detail; Sub-Saharan Africa and Nigeria. Especially attention is paid to what we have found concerning the deployment of information technology in developing countries, especially in Sub-Saharan Africa.

Based on the theoretical background presented in the previous chapters, Chapter 5 describes the research methodology. Different research methods are explained in detail and how the study was conducted concerning each method.

The rest of the chapters concentrate on the empirical results and their analysis, including contributions. Chapter 6 presents the results of different methods used when collecting the data. The results are analysed and summed up at the end of the chapter. Finally Chapter 7 provides discussion of the contributions based on the empirical results and literature review. In the conclusions, the results are evaluated in terms of validity, reliability, limitations, and further research.

1.8 Related work

Since I am writing a monograph, my approach is more 'personal' when compared with another option of the collection of papers written with co-authors. Since my work forms part of the research group, I will try to make a distinction to what is my contribution and what are the shared results. In this sense, I am using 'I', when I mean my personal goals or experiences, and 'we' when I am talking about the whole group. Many parts of this thesis are based on papers which were written by the whole group; the main author varies in the papers. During the project, our principle was to write all papers together as far as the paper concerns our shared research effort. In the text, I will refer to the corresponding paper, whenever there is one. So, in Appendix 1 I list the main publication (in order of publication year) I am referring, and indicate my contribution.

2 THEORETICAL FOUNDATION BASED ON INFORMATION SYSTEMS

This chapter examines the area of my research interest; information systems (IS) and information systems development (ISD), and the failure of ISD. First, the concepts of IS and ISD are defined, how the concepts are applied in this study. I also make a brief review of the history of information systems in terms of research and practice. ISD is reviewed concerning the related research, how ISD practices have been studied before, and what have been the contributions of these studies. According to my research questions, after we have clarified how ISs are developed in practice, the question of success or failure of ISD becomes relevant. We are especially interested in the risks of software development, as well as the question of how ISs sustainability in use can be analysed and improved during the ISD process. A literature review of these questions are summarised at the end of the chapter.

2.1 IS – research and practice

The term 'information system' (IS) has been used and defined broadly with different interests, thus the subject of IS is very *multi-disciplinary* or *inter-disciplinary*. There seems to be no *a priori* limit on the number of possible reference disciplines, so IS researchers need to remain receptive to a wide range of potential new perspectives on the issues with which they are concerned (Jones 1997).

Mingers and Stowell (1997) have edited a whole book concerning the question of whether IS is a discipline at all. First of all, IS is a relatively new area of study – about 20 years – including domains like computer and business systems, computing science, and management science. For being a discipline, there should be a general agreement about the domain of study, philosophical foundations, educational programmes, and appropriate research methods and methodologies. In the introduction they say that IS can be the discipline that concerns itself with the nexus of the varying domains (technology, information, mathematics, linguistics, semiotics, psychology, sociology etc.) as they bear upon the evolution of human communications and society. Thus we have to adopt a 'trans-disciplinary' approach to understand and influence the evolving nature and impact of technology upon society. This means that IS researchers should seek to do research which is good in terms of other disciplines, which is not an easy task to achieve (Jones 1997).

Anyway, even if the question of IS as a discipline is very interesting, I am not going to start to speculate what exactly is an IS discipline and how it should be defined. Neither am I going to justify whether IS is a discipline or not since I do not want to question that myself. Instead, I am going to understand IS and IS research as a multi-disciplinary subject, which is always characterised by the research interest (and context) in question.

2.1.1 Basic Definitions

Especially in North America the terms information system (IS) and management information system (MIS) are identical in meaning. They refer to the system providing technology-based information and communication services in an organization. Also a term like information management belongs to that group and it refers to the organization function that manages the system. The academic field may be termed like *information systems* (IS), management information systems (MIS), information management, or management of information systems (MoIS). The changes in terminology in the field reflect changes in the scope and consequently the research agenda during the history of IS. (Davis 2000)

Buckingham et al. (1987) define an information system as follows (according to Avison, 1997, p. 115):

> *A system which assembles, stores, processes, and delivers information relevant to an organisation (or to society) in such a way that the information is accessible and useful to those who wish to use it, including managers, staff, clients, and citizens. An information system is a human activity (social) system which may or may not involve computer systems.*

In information systems, technology is not the essence, even if the efficient management of information needs some modern information technologies. Land (1994) points out that information systems are essentially social systems, having information technology as one aspect. According to him, information systems exist in a real world which consists of objects (some concrete, such as machines; some abstract, such as budgets), people, rules (like established procedures), norms (like ways of thinking and practice), and commands (such as computer programs). Information technology applications are just one part of the means or instruments required in a given work activity. Thus the information system is not a synonym for a multi-user computer application, as it is often in day-to-day parlance (Korpela et al. 2000b).

In that point we can summarize that the information system is a social system (Walsham et al. 1990), having some technical aspects as well as social, political, and communicative aspects involved in work activity.

Examples of mainstream information systems are office automation systems and decision support systems. Examples of systems not considered as IS are computerized ignition systems and self-guiding missiles (Ein-Dor & Segev 1993). In this thesis I am not going to identify any further what kind of information systems there are and have been, what is the chronology of them. I can only mention that the development from early computation and data processing has been to more interactive, communicating, non-batch mode, user controlled, and natural language mode systems (c.f. Ein-Dor & Segev 1993). Later in this chapter, I will shortly examine the history of IS use.

For many researchers, IS development (ISD) has been the core of the field of IS (e.g. Iivari 1991, Hirschheim et al. 1995). It does not mean that other aspects, such as the role of IS use, are unimportant to the field, but there is a belief in the centrality of ISD. One simple reason for that is the truth that IS is an artefact that does not exist without IS development (Hirschheim et al. 1996).

Overall, *information systems development* is a specific kind of human activity. According to Mathiassen (1998), it is an intentional change process which is driven by certain more or less clear objectives. It includes phases such as analysis, design, programming, implementation, and maintenance, as well as – what we can call as supportive activities – like project management, quality assurance, and software process improvement. It also includes a group of actors that operate in a set of social and technical environments. The intentional changes focus on object systems or perspectives on computer-based information systems. That means the technical perspective focusing on the technical platform, the symbolic perspective focusing on the information contents, and the organizational perspective focusing on the use of the information system (Lyytinen 1987). The actual change process is characterized by several factors such as the experience and competence of the development group, the considered object systems, the dynamics of the objectives, and the social and technological environments in which the change process takes place (Mathiassen 1998). Also the applied systems development method plays an important role (Lyytinen 1987).

Usually an ISD process contains some programming or *software engineering* of the information-technological facilities which are not readily available. However, today ISD deals increasingly with adjusting and integrating prefabricated pieces of software to fit the needs of a specific work activity (cf. Lyytinen et al. 1998). In FIGURE 8, ISD includes some 'early design phase' tasks, as well as 'late design phase' tasks. In some cases, in the 'early design phase', it is realized that there are commercially available packages which can facilitate the target activity as required. In other cases the required information-technological means must be constructed from scratch. Thus software engineering and ISD are two closely related but separate fields of enquiry and practice. (Korpela et al. 2002b)

When speaking about ISD research or practice, we must distinguish between *approaches*, *methodologies*, and *methods*. In the first place, the field of IS can be identified by different *paradigms*. By paradigm I mean the fundamental set of assumptions adopted by a professional community which allow it to share similar perceptions and engage in commonly shared practices (Hirschheim & Klein 1992). Examples of different paradigms are positivism and interpretivism, to which I will come back to later in this chapter. In ISD *research*, what I understand by *research approach* is a concept composed by theories and practices which share some basic orientation about what is regarded as important in ISD. Research approach is a philosophy, a theoretical framework. Examples of research approaches are activity theory, structuration theory, and socio-technical theory, among others. (cf. Korpela et al. 2000b).

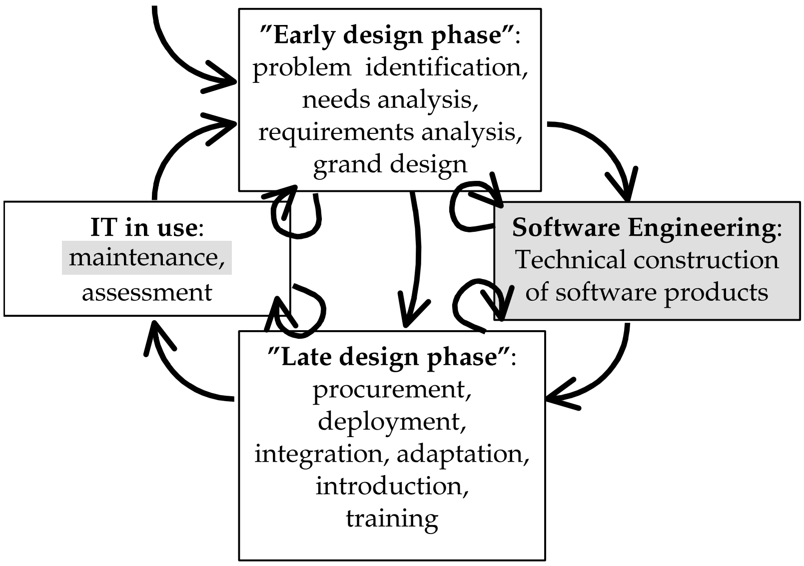


FIGURE 8 The roles of ISD (white) and Software Engineering (gray) in IS lifecycle (Korpela et al. 2002b)

Research methodology, on the other hand, means a systematic way of conducting ISD research by using a set of methods which are based on a shared, more or less explicit philosophy or approach (cf. Avison & Fitzgerald 1995). Mathiassen (1998) divides systems development research into three – what he calls – approaches: action research, practice studies, and experiments. All these methodologies, often in combination, contribute to the building of research-based knowledge in the form of theories and methods. Here the term methodology refers to the actual research method(s) used in a certain piece of research (Mingers 2001). Thus every study has its own individual methodology. Overall, in researching information systems development, field experiments, surveys, cases, and action research are particularly used as research methodologies (Avison 1997). *Method* is a way of conducting a specific task, like interview(s), observation or questionnaire(s). However, it can be difficult to precisely delineate the boundaries between method and methodology, or between methodology and a general research approach (Mingers 2001).

In ISD *practice*, ISD *approaches, methodologies, and methods* can be defined as was done with ISD research. Sometimes for example the same ISD methodologies can be applied to both research and practice. In ISD methodology we can see two distinct aspects (Hirschheim & Klein 1992):

process and framing. Process refers to the tasks, tools, and techniques used to accomplish the procedural component of a methodology. Framing refers to how the object system is perceived, and the types of changes which take place. Thus methodologies are shaping the process of ISD by improving it with a given framing. A systems development method is an organized collection of concepts, beliefs, values, and normative principles supported by material resources. Their purpose is to help the development group carry out the change process. As an example of ISD practical approaches we can mention the participatory approach, which has traditionally been strong in Scandinavia (e.g. Kensing & Blomberg 1998). Within the participatory approach, we can find two systematic methodologies: Cooperative Design (Greenbaum & Kyng 1991) and MUST (Kensing et al. 1998). Future workshop, which is a specific participatory design technique, is an example of a method. (Korpela et al. 2000b)

2.1.2 IS practice

If we like to examine IS practice historically, we need to consider the challenges both in the use of IS as well as in the IS development. The challenges faced in IS practice have changed with new technologies and new types of applications that have been requested. To view this history I firstly refer to Mathiassen (1998), who collected systems development challenges in three eras by referring Avison et al. (1988), Lyytinen (1989), and Applegate et al. (1996). According to Mathiassen the history can be divided into three eras, which cover (I) the early sixties to the mid seventies, (II) the mid seventies to the late eighties, and (III) the late eighties to the late nineties. Each era introduces new trends which lead to additional challenges. Still, most of the previous trends remain sustainable.

The first period in IS practice covers early sixties to the mid seventies. At that time systems development was mainly practiced as a technical discipline. IT was used to automate the existing manual processes and each application was treated as a separate entity. The overall purpose of using IT was to increase productivity and efficiency. The technology applied was mainframes, batch processing, and databases. Developers were basically programmers and the dominating strategies for improving practices were based on introduction of new tools for programming and new methods that enforced the basic project management disciplines. (Mathiassen 1998)

The second period covers the mid seventies to the late eighties. At that time there were at least two important trends. First, distributed, more user friendly, and more readily available technologies created the basis for new and more extensive uses of IT. Secondly, the dominating position of technical skills was complemented with end-user computing and collaboration between different professional groups. At the same time there was a shift away from technology towards its use. The key words in technology were distribution, PC's, local networks, graphics, and expert systems. Analysis, design, and collaboration were emphasized as key activities or skills and quality assurance became a key factor in improving practices. Applications were now being developed to support professional work (including CASE tools), many systems became highly integrated, and there was a considerable expansion of the use of IT in other industrial artefacts. (Mathiassen 1998)

The third period covers the late eighties until the change of the millennium. The most dominant change during this time has been the global networks which provide for transcending conventional boundaries for using IT. Applications have now become a more integral part of business strategies and they have created new opportunities to establish alliances and collaboration across organizational and national boundaries. At the same time, still more types of standard application software are used challenging the established tradition of developing unique applications tailored to specific organizational needs. Thus the technological key words are global networks, multimedia, mobile computing, and standard software. During this era there has been an increased focus on skills related to specific business domains and on architectural skills needed to utilize networks and adapt and integrate applications. The new tradition for improving practices is based on a concern for systems development processes and for the technical and cultural environments that shape them. (Mathiassen 1998)

Friedman (1989) evaluates the development of computer systems for business applications, which began in the mid-1950s. The first computers were built in the 1940s. The first phase starts from the beginning of computerizing until the mid-1960s, and the second phase from the mid-60s until the beginning of the 1980s. The third phase involves the 1980s. According to him, the fourth phase was about to begin at the time of his book. He approaches the development phases a little bit differently, compared to Mathiassen (1998), focusing on constraints in each time era. He argues that the three phases have been dominated by constraints on further computerization. He specifies the following constraints for each phase (Friedman 1989, p. 59):

1. Phase 1, hardware constraints: hardware costs and limitations of capacity and reliability.
2. Phase 2, software constraints: productivity of systems developers, difficulties of delivering reliable systems on time and within budget.
3. Phase 3, user relations constraints: system quality problems arising from inadequate perception of user demands and inadequate servicing of their needs.

During the first phase, the cost and capacity of computer hardware was the major constraint on the spread of computerization. The essence of the second phase was that systems development or software costs and performance came to displace hardware costs and performance as the major constraint on the spread of computerization. Thus the emphasis was on the technical issues, in order to turn to the user relations constraints in the 1980s. Software productivity did not rise dramatically, as hardware productivity did during phase one and later on. Also, user needs became increasingly difficult to specify, and the user relations problems became more important to solve because users were becoming more knowledgeable and powerful. That was due to the spread of PCs and a general increase in computer awareness during the 1980s. 'The 1980s could go down as the decade of the user', declared an editorial in the UK computer press in 1986 (According to Friedman 1989:

Computer News, July 17). Friedman suggests a future scenario of computer systems development within user organization (end-user computing) to be compared with the development of inter-organizational systems. He would label this fourth phase the phase of organization environment constraints. He identifies the causes of the constraints as follows (Friedman 1989, p. 337):

1. The problem of accurately identifying and accessing strategic information, particularly about hostile and potentially hostile agents.
2. The problem of establishing sufficiently cooperative relations with complementary organizations.
3. The problem of reducing supplier, and particularly consumer, resistance to operating through computer-mediated communication.
4. The problem of agreeing and enforcing standards and protocols both for systems development and for use with cooperating organizations and customers.
5. The problem of security, particularly as such systems become much more vulnerable to hackers and thereby to industrial espionage.

Friedman's prediction seems to match many ways with Mathiassen's view of the late twentieth century, especially concerning the strategic and collaborative systems, global networks, and architecture. All we have personally experienced nowadays is the problem with security, especially concerning viruses. TABLE 1 illustrates Mathiassen's eras with Friedman's constraints.

Probert (1997) describes the historical nature of IS practices as being moved from technological questions to business needs and further to more strategic views of possibilities for exploiting the technology. Nowadays the employees can have a vision of what the organization is trying to achieve and there are shared values. There is a need for a shift from management to leadership. The role of IT changes from 'solution' to 'enabler' and this has implications for IS as well. Firstly, the role of people is enhanced from users to utilisers, and secondly, the boundary between people and the IT they are interacting with becomes difficult to set (concluding to definition of information system above). Also, the implementation of information systems can alter the distribution of power within organizations.

The organizational environment is continually changing nowadays adjusting the commercial technology, particularly the demands of e-business, the globalization of markets, and the pressure for reengineered, quality-oriented organizations (Russo 2000: Truex et al. 1999). Thus organizations are moving from functional, hierarchal management structures to team-based, networked structures. Strategic alliances and integration across the supply chain are becoming usual and IT enables these changes. Furthermore, IS is not only supporting the organization, it can be the organization, like in the case of Amazon.com. (Russo 2000)

The work of ISD is changing accordingly. The focus is on small-scale, rapid development in shortened time horizons. The developers are also faced with a growing number of platforms, homogenous or heterogeneous networks, and component-based development as well as Web development. The required skills of systems developers include teamwork skills, adaptability, business knowledge, multi-media skills, and telecommunication skills. (Russo 2000)

TABLE 1 Systems development challenges and constraints

	Era I: 60s – mid 70s	Era II: mid 70s – late 80s	Era III: late 80s ~ 2000	
Purpose	Productivity and efficiency	Individual and group effectiveness	Strategic and collaborative	
Application	Automation Separate systems	Support Integrated systems Embedded systems	Strategic systems Process integration Collaboration	
Technology	Mainframes Batch processing Simple terminals Databases	Distribution PC's Local networks Graphics Expert systems	Global networks Multimedia Mobile computing Standard software	
Skills	Programming Management	Analysis Design Collaboration	Domain Architectural	
Improvement	Methods and tools Project management	Quality assurance CASE	Process improvement	
Constraints	Hardware	Software	User relations	Organization environment

2.1.3 Information Systems Research

Information system research is a multi-disciplinary topic and very much a social, rather than an entirely technical science. Foundations can be found in philosophy and in the organisational and behavioural sciences, as well as in mathematics and the natural sciences. (Galliers 1994)

Looking back at the history of information systems research, the main questions at the beginning seemed to be in the foundation of new discipline, diversity in research problems, the pluralism of methods, reference disciplines, and theoretical foundations, and missing cumulative tradition. The distinction of IS research between North America and Europe was emphasized more in the early years, but nowadays the discipline has become more integrated approving the diversity of the area.

Diversity in Paradigms. Generally, many researchers categorize the IS research into two paradigms – positivist and interpretive approaches. A useful way of categorizing paradigms is by looking at their underlying *ontological* and *epistemological* assumptions. Ontological assumptions are concerned with the beliefs about 'reality' and epistemological assumptions with beliefs about what counts as knowledge and how it can be obtained (Myers 1997, Klein & Myers 1995). According to Hirschheim (1985), knowledge is not infallible but conditional; it is a societal convention and is relative to both time and place.

In the United States the approaches in research were mostly based on natural science, which is a fairly strict conception of science. In this positivists approach, an epistemological assumption posits beliefs (emerging from the search from regularity and causal relationships) and scrutinizes them through empirical testing. The basic ontological assumption of positivism is that reality is objectively given and can be described by measurable properties which are independent of the observer (researcher) and his or her instruments (Myers 1997). Positivism has a long and rich historical tradition (back to 17th century, see Hirschheim 1985). In the beginning of IS research traditions, positivism was thought to be so embedded in our society that the research results not grounded in positivist were not considered as scientific and therefore invalid (Hirschheim 1985). In 1980s, 97% of the IS literature fell under positivist epistemology (Orlikowski & Baroudi 1991). According to Hirschheim (1985, p. 16) positivism is based on five pillars:

1. Unity of the scientific method
2. Search for causal relationships
3. Belief in empiricism
4. Science (and its process) is value-free
5. The foundation of science is based on logic and mathematics.

The strength of the positivist approach lies in its ability to provide a wide coverage of various situations and to be fast and economical, as well as in its rigour and replicability in the conduct of science research. (Avison 1997)

The limitations include for example that the quest for universal law leads to a disregard for historical and contextual conditions as possible triggers of events or influences on human action. The design and use of IS in organizations is embedded in a social context, marked by time, locale, politics, and culture. Neglecting these influences may reveal an incomplete picture of IS phenomena. Due to the positivist research perspective it tends to disregard the historical context of phenomena, positivist research studies are rooted in the status quo. Likewise, the positivist aims to explain and predict external reality which implies that people are not active makers of their physical and social reality. (Orlikowski & Baroudi 1991)

In Europe, IS academics relied more on *interpretive studies*. Within interpretive studies, social and organizational issues concerning IS were increasingly recognised, seeking to obtain in-depth understanding of IS phenomena. The most widely used techniques were (and are) interviews, observation (field experiments), surveys, case studies, and action research. (Avison 1997)

Interpretivism asserts that reality, as well as our knowledge thereof, are social products and hence incapable of being understood independent of the social actors (including researchers) that construct and make sense of that reality. Social constructions include such elements as a language, consciousness, shared meanings, and instruments (Myers 1997). Ontologically, the interpretive perspective emphasizes the importance of subjective meanings and social-political as well as symbolic action in the processes through which humans construct and reconstruct their reality (Orlikowski & Baroudi 1991: Morgan 1983). Interpretive information systems research assumes that the social world (that is, social relations, organizations, division of labour) are not ‘given’. Rather, the social world is produced and reinforced by humans through their action and interaction (Orlikowski & Baroudi 1991).

The main strength of the interpretive approach is its ability to look at change processes over time, to understand actors' meanings, to adjust to new issues and theories as they emerge, and to contribute to the evolution of new theories (Avison 1997: Easterby-Smith et al. 1991).

There are also some criticisms. First, the interpretive perspective does not examine the conditions, often external, which give rise to certain meanings and experiences. Second, research in this perspective omits to explain the unintended consequences of action, which by definition cannot be explained by reference to the intensions of the humans concerned. These unintended consequences of action are often a significant force in shaping social reality. Third, the interpretive perspective does not address structural conflicts within society and organizations, and ignores contradictions which may be endemic to social systems. Finally, the interpretive perspective neglects to explain historical change: that is how a particular social order came to be what it is, and how it is likely to vary over time. (Orlikowski & Baroudi 1991)

Nowadays both positivist and interpretive approaches are mutually accepted and relevant to aspects of IS research since the nature of IS discipline concerns both the social and technical aspects of organizations (Probert 1997). Still, there would appear to be some informal evidence to suggest that European IS research has a tendency to emphasise relevance, while that of the USA has a tendency to emphasise rigour (Avison 1997, Galliers 1997). I would say that both these two approaches can be applied to supply and support each other, to make results more reliable. Of course, it is not always possible to do that. TABLE 2 summarises the epistemological differences between positivism and interpretivism.

Orlikowski and Baroudi (1991) identified a third paradigm, besides positivism and interpretivism, namely the *critical philosophy*. The critical research attempts to critically evaluate and transform the social reality under investigation. Where the other two are content to predict or explain the status quo, the critical perspective is concerned with criticizing existing social systems and revealing any contradictions and conflicts that may inhere within their structures. The central idea within critical philosophy is the belief that social reality is historically constituted and hence human beings, organizations, and societies are not confined to existing in a particular state (Orlikowski & Baroudi 1991: Chua 1986). Another important idea in critical philosophy is that things can never be treated as isolated elements, but as totality. This dialectical relationship between elements and the totality is understood to be shaped by historical and contextual conditions. (Orlikowski & Baroudi 1991)

Social reality is understood to be produced and reproduced by humans, but also as possessing objective properties which tend to dominate human experience. Due to the dialectical understanding of elements of the whole, as well as the belief in human potentiality, the critical research philosophy emphasizes the processual development of phenomena. Social relations are not posited as stable and orderly, but as constantly undergoing change. This instability is conceptualised in terms of fundamental contradictions that inhere to the social relationships and practices of societies and organizations. (Orlikowski & Baroudi 1991)

TABLE 2 Differences between positivist and interpretivist epistemology (Myers 1997, p. 243)

Epistemological assumptions of positivism	Epistemological assumptions of interpretivism
1. Experience is taken to be objective, testable, and independent of theoretical explanation	1. Data is not detachable from theory, for what counts as data is determined in the light of some theoretical interpretations, and facts themselves have to be reconstructed in the light of interpretation
2. Theories are held to be artificial constructions or models, yielding explanation in the sense of a logic of hypothetico-deduction (if IT is true, phenomenon x follows)	2. In the human sciences theories are mimetic reconstructions of the facts themselves, and the criterion of a good theory is understanding of meanings and intentions rather than deductive explanation
3. Law-like relations (generalisations) are derived from experience and are independent of the investigator, his/her mehods and the object of study	3. The generalisation derived from experience are dependent upon the researcher, his/her methods and the interactions with the subject of study. The validity of the generalisations does not depend upon statistical inference 'but on the plausibility and cogency of the logical reasoning used in describing the results from the cases, and in drawing conclusions from them' (Walsham, 1993, p. 15)
4. The language of science can be exact, formalisable and literal	4. The languages of the human sciences are irreducibly equivocal (because of multiple, emergent meanings) and continually adapt themselves to changing circumstances
5. Meanings are separate from facts	5. Meanings in the human sciences are what constitute the facts, for data consist of documents, intentional behaviour, social rules, human artefacts, etc., and these are inseparable from their meanings for agents

Critical studies tend to be longitudinal. The research methods of choice are long-term historical studies and ethnographic studies of organizational processes and structures. Quantitative data collection and analysis are used, although to a lesser extent. Critical researchers depart from their interpretive colleagues, in that they believe interpretation of the social world is not enough. The material conditions of domination also needs to be understood and critiqued, and these are typically not accessible by merely asking participants who often are unable to perceive and penetrate the circumstances that shape and constrain them. (Orlikowski & Baroudi 1991)

The critical research philosophy towards the relationship between theory and practice is that the role of the researcher is to bring to consciousness the restrictive conditions of the status quo, thereby initiating change in the social relations and practices, and helping to eliminate the bases of alienation and domination. (Orlikowski & Baroudi 1991)

The critical approach has some weaknesses. For example, the socio-economic class is seen as the primary determinant of antagonistic social relations. This almost exclusive focus on economic factors obscures the importance of other factors such as race and gender that have also led to dominating and repressive social relations. This selectivity of the perspective is also reflected in the critical researchers' recognition of the inherent opposition or contradiction in social relations. Further, critical researchers are not reflexive, not applying their notions of transcendence to themselves, and hence not accomplishing self-transformation or praxis. Finally, the form of theory and knowledge in this tradition is uncertain. (Orlikowski & Baroudi 1991)

Robey (1996) emphasises the philosophical position of extreme *pragmatism*, justifiable in part because of the IS's close alliance with the practical problems of business. Theories and research methods are intellectual artefacts, produced by scholarly communities with some purpose in mind. Discipline is required to prevent methodological pluralism from becoming methodological anarchy. Theoretical foundations for research and specific research methods are justified by research aims, or purposes. They should not be chosen because they conform to a dominant paradigm or because the researcher believes in their intrinsic value. Rather, theories and methods are justified on pragmatic grounds as appropriate tools for accomplishing research aims. Such discipline requires that researchers be clear about their aims, that they justify their choices of theory and method, and that they maintain a balance among research aim(s), theoretical foundation, and research methods. (Robey 1996)

Also Wicks and Freeman (1998) offer pragmatism, or *new pragmatism*, to increase the relevance of research and to provide room for ethics. They discuss the relative absence of ethics in both positivism and anti-positivism epistemology, and how researches should go beyond that debate and develop research that is focused on serving human purposes. They offer an approach to organization studies which they term 'pragmatic experimentation'. In this approach the focus is on the practical relevance as well as the desire to search for innovative approaches ('experimentation'). Ethics has a central role.

Historical Reasons for Diversity. The diversity in research paradigms seem to reflect in the history of IS research and its development. This diversity can be seen in the problems that researchers address within the discipline, in the theoretical foundations and reference disciplines, and in the methods used when collecting, analysing or interpreting data. Benbasat and Weber (1996) separate the development of IS research in the USA and Europe. I would add Scandinavian research as a separate or additional tradition included in the European tradition.

Briefly, in 1977, in North America, the initial major research journal of IS, the *MIS Quarterly*, was published. The journal provided a focal point for researchers who were interested in publishing IS research. Authors could be reasonably confident that their papers would be read by colleagues who understood and supported their interest, not to doubt the importance. The second key event was in 1980, when the first *International Conference on Information Systems* (ICIS) took place. The major objective of the ICIS's founders was to establish a forum for presenting good IS research. That was accomplished in two ways: 1. through focusing on and celebrating high-quality

IS research, and 2. through establishing a doctoral consortium designed to influence the training of future IS researchers and ultimately the quality of IS research. Benbasat and Weber (1996) consider that the first ICIS conference had a lasting impact on the IS discipline. It shaped the agenda of the North-American IS research for the next decade. Its singular outcome was the reification of reference disciplines in providing the foundations for IS research. During the 1980s, the IS discipline's focus shifted from discussing *what* to study to *how* to study. IS literature increasingly emphasized approaches in undertaking research, including the selection of appropriate designs, measurement issues, statistical analysis, and epistemology (Benbasat & Weber 1996).

The idea of the European view was that North American research was based mainly on an orthodox, positivistic tradition, and that it made exclusive use of 'objective' empirical research approaches. It was also claimed to advocate one universal scientific method. Many European academics, on the contrary, advocated greater pluralism, more diversity, greater use of methods that allow researchers the scope for interpretation, and adoption of theoretical perspectives that are not founded on a rational and mechanistic view of the world. The researchers sought to discredit the perceived hegemony of the orthodox model, meaning the North American research model. (Benbasat & Weber 1996)

There are two meetings that perhaps best characterize the European view of IS research during 1980s. They were IFIP WG 8.2 (The Interaction of Information Systems and the Organization, established in 1977) conferences that were held in Manchester in 1984 (Mumford et al. 1985) and Copenhagen in 1990 (Nissen et al. 1991).

In 1984, in the IFIP WG 8.2 *Manchester* conference, the purpose was to discuss alternative research approaches in IS research. Fitzgerald et al. (1985) notes that the interest of working group 8.2 lies somewhere in between the two extremes, some of its concerns are technical in nature and some more social. This might be thought to argue strongly for an acceptance of a pluralism of methods in this area of research. Their concern was that the scientific viewpoint of IS research originated particularly from computer science, and that the acceptance of pluralism of methods is thus unlikely. The other concern was that universities and research committees do not support pluralism of research methods, instead they still value more traditional scientific values.

Despite the traditional values, the Manchester conference was seen as a phase in which to take a step forward in the direction of defining IS as something different to computer science or engineering. Bjørn-Andersen (1985) concludes that IS must be seen as a social science discipline and the academics should be closer to understanding the research field in its own right, and developing theories, methodologies, and tools relevant to addressing the problems. The other conclusion was the agreement of methodological pluralism to support the effectiveness in research. In addition, the researchers cannot just be concerned with the problem of effectiveness, they must also ask what their research is used for. Thus he started to talk about new ethics for IS research.

The IFIP WG 8.2 working conference in *Copenhagen* in 1990 followed the one in Manchester in 1984, where the discussion of alternative research approaches had begun. The Copenhagen conference was thus an attempt to take stock of and then critically assess the various research approaches that were being used by people in the IS community. In the call for papers it was said that at the Manchester Symposium some of the documented methods (like phenomenology, contextualism, trade union perspective, and critical theory, see later in this chapter) have influenced recent research projects, but some of these do not meet customary standards of rigor and validity. This raises serious concerns in the research community and others affected by the continuing deployment of information technology. The Copenhagen conference continued the discussion and it was needed because new research approaches had surfaced. These new approaches apply insights to the information systems use and development from ethnography, anthropology, philosophy of language, and other reference disciplines. (Klein et al. 1991)

The Copenhagen conference was organized around the assumption that 'radically different' research approaches do exist in the information systems research community (Klein et al. 1991). The discussion of the 'power' of traditional approaches and 'scientific methods' in Manchester 1984 had been 'won'. Researchers were not refused grants because they favour action research or prefer qualitative to quantitative methods. Instead of 'radically different' research approaches, IS research still remains something of a fashion industry with the choice of fashion being influenced more by the prejudices of governments and funding bodies than by the needs of the community (Mumford 1991).

The fact that three research methods – surveys, laboratory experiments, and case studies – still account for almost 90% of the published material suggests that the most commonly held attitude in the ISR community is not methodological pluralism, but the belief in the supremacy of a particular set of methodological postulates favouring certain methods and discriminating against others. (Klein et al. 1991)

The Manchester and Copenhagen meetings were milestones in the effort to inaugurate additional research approaches needed to explain and understand information systems. These meetings have shown the justification and legitimacy of qualitative researchers' approaches. The next meeting to review qualitative research approaches was organized in *Philadelphia* in 1997, having the title 'Information Systems and Qualitative Research'. At that conference there was an emerging acceptance of diversity in research approaches. Whereas the term 'qualitative' once carried the connotation of 'anti-positivist', in Philadelphia there was qualitative research that drew confidently upon positivism or other forms of deductivist approaches. There were also much evaluations of qualitative research. A distinguishing feature was that the authors were deliberately reflecting on the accomplishments of qualitative IS research since the times of Manchester and Copenhagen. Thus the purpose of the Philadelphia meeting was purposely self-reflective and evaluative. (Lee & Liebenau 1997)

In the year 2000 IFIP WG 8.2 conference in *Aalborg, Denmark*, the title was

'IS2000: The Social and Organizational Perspective on Research and Practice in Information Technology'. This IS2000 working conference was intended as a WG 8.2 milestone, the turn of the first century of the information age. The key question was: 'What is the status of the discipline of information systems as we stand at the juncture of the new century?' So, what is it? Baskerville and Stage (2000) conclude based on the presented papers that for example significant and irreversible shifts in social theory have not appeared, information technology is still central in the social interaction in organizations, and indeed has become more critical because the traditional mutual benefits between employers and employees has become more fluid. Human language barriers remain the fundamental roadblocks to IT implementation. In information systems development the chancing mix of skills and knowledge is required.

According to Baskerville and Stage (2000), many traditional problems have been solved, like finding acceptable ways of evaluating the organizational worth of social elements, developing better research methods, and linking our discoveries into a system. Solutions are for instance developing hybrid methods that link research and practice, or centering processes for integrating theories. There are also new problems like the lack of integrated approaches to risk management, the social costs being unleashed on people by the wiring of society, and our headlong rush into globalized systems without a proper understanding.

There has been an incredible evolution in the relationship and exchanges between human beings and computing machinery. Computers have become inscribed socially with autonomy and human values, and organizations have become hybrid collectives of humans and non-humans. Consequently we need new norms and methods for defining and studying the relationships and exchanges between human beings and computers. Information technology has not only broken through technical barriers, but through social ones as well. For example, the web design implies changing an incredible web of social relationships. As Baskerville and Stage (2000) say, the new century raises on our horizon other complicated issues like the *ethics* of these new IS and IT concepts and practices. Thus the ethical questions rise again after the paradigmatic debate. (Baskerville & Stage 2000)

The theme of the social and organizational perspective was presented also in *Boise*, USA, in 2001. The overall theme of the conference was 'Realing Research and Practice in Information Systems Development: The Social and Organizational Perspective'. The conference sought to understand the impact of information systems development and use on society, organizations, and individuals. The papers attempted to help researchers understand and address the current state of information technology in the world today, instead of technical questions. (Russo & Fitzgerald 2001)

Examples of qualitative research approaches in WG 8.2. conferences. The Manchester conference offered a presentation to some new IS research approaches like phenomenology (Boland 1985), trade union perspective (Sandberg 1985, see 'Scandinavian tradition' later in this chapter), contextualism (Pettigrew 1985), and critical theory (Lyytinen & Klein 1985).

Phenomenology emphasises the methodological study of consciousness in order to understand the essence and meaning of experience, *what* things are. It sees itself as an opposite to positive science, which is concerned with finding out *how* things work. The central idea of phenomenological *hermeneutic* is that the use, design, and study of information systems can be read (understand a text) and interpreted by people other than its authors. As an example Boland (1985) gives a study where he conducted a series of in-depth interviews with a system designer (for four years). The result was a series of brief statements describing the essence of the experience of communicating with others during the design, the experience of moving through the organizational space, and the experience of moral choice in the design process. Boland (1991) continues to discuss hermeneutics approach in Copenhagen. Viewing the information system use as a hermeneutic process can help us understand how systems outputs are made meaningful by their readers and how they become incorporated in decision and action.

Critical theory (Lyytinen & Klein 1985), based on Habermas's (1984, 1987) writings, suggests that information systems which are designed to increase organizational effectiveness must also increase human understanding and emancipate people from undesirable social and physical constraints, distorted communication and misapplied power. People's behaviour is a derived form of their 'knowledge interests', which can be technical, practical or emancipatory. Critical theory or critical social theory (see later in this chapter) emphasises the role of interpretive and hermeneutics methods in research which has a social context (Lyytinen & Klein 1985).

In *contextual approach* (Pettigrew 1985) there are three basic elements to an analysis: the process component, the context component, and the outcome component of the process under investigation. Contextuality seeks to analyse processes in their intra organizational and social, economic, political, and business context.

In the Copenhagen conference some new research directions were also introduced like semiotics (Andersen 1991), activity theory (Kuutti 1991, Bødker 1991b, see Scandinavian Tradition later in this chapter), Ihde's phenomenology (Rathswohl 1991), and postmodern thought, as well as significant changes in the direction of others such as cooperative system design (see Scandinavian Tradition later) (Klein et al. 1991).

The *postmodern thought* means pluralism in IS research methodologies, a broader perspective, anthropological-historical method, and emphasis on quality rather than on the quantity of research publications (Achterberg et al. 1991).

The *semiotic approach* focus is on people's own interpretations of what they do and it can be investigated by carefully studying recordings of their communications during work. A key assumption is that IS should fruitfully be viewed as media for social interaction and not as models of reality. From the study of language and how it relates to work situations design ideas emerge from discerning repeating patterns, concepts, and metaphors in the ways users talk about their work. The results of this analysis deepens the designers' understanding of the work they are designing for and can be a basis for making users' conscious of concepts and attitudes which they are not aware of in daily work. (Andersen 1991)

Ihde's phenomenology emphasises how to conceptualize and describe the human use of technology. Briefly, Ihde argues that technology mediates human experience and that there are general principles that describe technology-mediated experience. He develops the idea that technology mediates the human-world relation with three elements (human, world, and instrument), which can be understood in two ways: as embodiment relations (where the world is directly experienced by the human through the use of the tool), and as hermeneutic relations (where the instrument serves as a representation of the world and it is only through the instrument that an event is experienced). (Rathswohl 1991)

Grounded theory (Calloway & Ariav 1991, Toraskar 1991) was also discussed in Manchester. According to Järvinen (1999) the basic theme in grounded theory is the discovery of theory from data systematically obtained from social research (Glaser & Strauss 1967). A grounded theory is discovered, developed, and provisionally verified through systematic data collection and analysis of data pertaining to the phenomenon under study. A well-constructed grounded theory will meet four central criteria for judging the applicability of theory to a phenomenon: 1) fit, 2) understanding, 3) generality, and 4) control. In Manchester Calloway and Ariav (1991) described how they developed and used a qualitative methodology in order to study relationship among designers and tools by using the grounded theory. Toraskar (1991) used the grounded theory in study to show how managerial users evaluate their decision-support.

Scandinavian Tradition. Approaches where a researcher actively associates him/herself with a particular practical outcome of the research tended to come from the Scandinavian countries (Galliers 1985). The characteristics of the Scandinavian IS research approaches are plurality in theories, research approaches, topics, and outcomes. One distinctive feature has been its strong conceptual and philosophical orientation. Compared to the North American MIS tradition, the Scandinavian approaches emphasize IS evolution, user-participation, alternative process models, seek varying and innovative theoretical foundations for IS and ISD, and apply dominantly anti-positivistic and action oriented research approaches. The early and innovative start in the 1960's, the socio-political history, and dynamics of the Scandinavian societies combined with a rapid and intense utilization of computers formed a fertile environment for the establishment of the '*Scandinavian school*'. The important forums for communication between the Scandinavian research community have been annual meetings called *IRIS* (Information systems Research in Scandinavia) which was already launched in 1978, and the Scandinavian Journal of Information Systems since 1988. Iivari and Lyytinen (1998) draw a map of an intellectual community of Scandinavian approaches from the 1960's of the Langeforsian beginning to the early 90's, concentrating on the research of ISD. As Iivari and Lyytinen point out, the map is not complete but provides pointers to the major contributions.

The Langeforsian beginning (Langefors 1966) refers to the *infological* *approach*. The basic idea of the infological approach is the distinction between the infological problem of defining the information to be provided by the system in order to satisfy the needs of its users and the datalogical problem of designing the structure and operation of the system and exploiting current information technology (Langefors 1974). The purpose of infological concepts was to develop a set of concepts for solving IS design problems distinct from the implementation (programming) problems by freeing them from all aspects of the computer technology and to make them more meaningful to users and designers. The infological tradition has a role as the root of most of the other approaches. There are some contributions that continues Langefors' work. The first one was Langefors and Sundgren's (Sundgren 1973) work on the conceptual / infological modelling in the shift of the 1960's and 1970's. The other was the development of ISAC method for information analysis (Lundeberg & Andersen 1974). ISAC was developed in a number of action research projects and was thus one of the first methods which incorporated an attempt for empirical validation (Lundeberg 1976). (Iivari & Lyytinen 1998)

Formal approaches can be regarded as an attempt to extend Langeforsian ideas (in 1980s and 1990s). This has resulted in several more rigorous approaches focusing on IS analysis and design problems that lend themselves for formal manipulation, possible to be carried out by computers. The formal approaches to systems design in Scandinavia cover two research streams: research on Computer Aided Information Systems Design (Bubenko et al. 1971) or Computer Aided Systems / Software Engineering (CASE), and research on Conceptual Information Modelling (CIM). (Iivari & Lyytinen 1998)

In the *sociocybernetic (SC) approach* IS design is analysed in terms of human action (since 1970s), formalising it using the sociocybernetic theory of acts (Aulin-Ahmavaara 1977) and information economics (Marschak 1974). IS design is viewed as an inquiry process – a set of acts to obtain knowledge about the IS domain – which supports a steering committee to decide upon the major alternatives concerning both the IS product and the ISD process. The IS design act is further conceptualised either as observation / analysis acts or manipulation / refinement acts. The IS design process transpires an execution of sequential and parallel IS design acts, each act increasing the knowledge about the IS design situation and / or refining the IS artefact to be designed. In the SC approach flit concepts like meta IS design act, two layered abstractions, organizational, language, and technical context, multi-dimensional process model, just to mention a few. The practical use of the SC approach has been modest, one reason is its complexity which results from its generality. The theoretical basis of the model is also eclectic, complex and loaded with difficult vocabulary. (Iivari & Lyytinen 1998)

The *language action (LA) approach* in Scandinavia (mostly in 1980s) to a large extent emerged as an evolutionary step and a radical reinterpretation attempt of the infological tradition. Goldkuhl and Lyytinen (1982, 1984) wrote several articles that established the theoretical basis for the LA approach. The basis and theory of LA is quite complex having the philosophy of language, concepts of universal pragmatics, language games, Habermas' theory of knowledge interest and the theory of communicative action in it. The SA theory suggests that speech acts form basic units of communication and always express human intent, such as making a promise or asserting a claim. This applies also to communications in the IS. Advocates of the LA approach assume that the task of the IS developer is to recognise all types of speech acts explicitly and to use this knowledge in the design of the system. The LA approach consists of a rich set of concepts for the analysis of IS as instances of intricate social communications. At the same time its weakness is in its complexity. (Iivari & Lyytinen 1998)

Scandinavia has also been influential in the emergence of *object-oriented* languages – for example Simula 67 (Dahl & Nygaard 1966), Delta (Holbæk-Hanssen et al. 1976), and Beta (Madsen et al. 1993) – and analysis and design methods – like OCTOBUS (Awad et al. 1996), OOram (Reenskaug et al. 1995) just to mention a few. Jacobson's OOSE method has received the most international recognition (Jacobson et al 1992). OOSE is based on two distinctive ideas: the concept of '*use case*' and the division of analysis objects into entity, interface, and control objects. Compared to most methods of OO analysis and design, OOSE is user-oriented. Object-Oriented Business Process Re-engineering of Jacobson et al. (1995) applies a conceptual structure analogical to OOSE to model organizations. (Iivari & Lyytinen 1998)

The basis for the *structuration theory* (ST) is on Gidden's theory of structuration (Giddens 1979, 1984, 1990). The approach has raised interest since the mid 1980s in the US (e.g. Orlikowski 1995), in the UK (Walsham 1993) as well as in Scandinavia (Lyytinen & Ngwenyama 1992, Karsten 2000, Käkölä 1996). Recently ST has become one of the main strands of 'qualitative', interpretivist mode of research in the IS community. The main concern of ST is to shift the social theory from epistemology (what type of theorizing is appropriate to understand the social domains) to ontology (what a social domain is and how it is structured). The basic principle of the structuration theory is that dualisms – action/structure, individual/society, determinism/ voluntarism, technology/social – should be conceived as dualities, as two sides of the same coin. The concept of duality formulates a complex conceptual apparatus and terminology to analyse how different forms and levels of the social structure emerge from practices. A strength of the ST approach is its theoretical depth and richness, though abstract vocabulary, and strong focus on ethnographic research methods. The weakness is in its generality and the difficulty to interpret and use the results. (Iivari & Lyytinen 1998)

Among the most influential approaches concerning my research interest was the *Socio-technical design* (STD), followed by the trade union approach and participatory design. When STD is applied to ISD, it can be seen as an attempt to extend the Langeforsian ideas into topics that deal with work design (mostly in 1970s). Two major inspirations can be identified for the development of socio-technical ISD approaches in Scandinavia. These are the Norwegian Industrial Democracy Project in the 1960's (Thorsrud & Emery 1969) and the impact of ETHICS method for Scandinavian Research (Mumford & Ward 1968, Mumford & Hensall 1979, Mumford & Weir 1979). The experiences to test STD in practice encountered severe problems during implementation (e.g. Ehn 1988, Bjørn-Andersen et al. 1979). These lead to a more radical trade-unionist approach in the early 1970's. (Iivari & Lyytinen 1998)

The most distinctive features in the *trade union (TU) approach* include its origin, the Scandinavian industrial relations as its institutional background, the role of trade unions as sponsors and partners of its major research projects, and its belief in trade unions as legitimate representatives of the resources weakest groups (from 1970s to 1990s). The TU approach can be regarded as an attempt to seek a radical antithesis to the infological and Socio-technical design approaches that were viewed as management oriented, considering organizations as harmonious assemblages of people, and were largely value-neutral and positivistic in their world-view. The concern was the ways in which computer-based systems were introduced into the workplace and to the deleterious effect these systems were having on workers. The workers and their unions were concerned that the introduction of computers would reduce their control over their immediate work situation as well as the overall planning and administration of production (Kensing & Blomberg 1998). Iivari and Lyytinen (1998) identify three major generations in the evolution of the TU approach. The first generation covers the first three major projects, NJMF in Norway (Nygaard 1979), DEMOS in Sweden (Ehn & Sanberg 1979), and the Danish DUE project (Kyng & Mathiassen 1982). The second generation covers the UTOPIA (Ehn 1989) project which aimed to promote industrial democracy and the quality of work and product. The hope was that computer tools and environments could be prototyped and built that would strengthen the position of labour in their efforts to improve working conditions and the quality of working life (Kensing & Blomberg 1998). The third generation emphasises *cooperative design* (e.g. Kyng & Greenbaum 1991), where users and system designers cooperate and contribute each to the systems development process. The main contributions of the TU approach comprise of its critical value orientation challenging the management to determine the goals of the system development. So, ISD may be governed by alternative values. The radical edge of the TU approach has been weakening. (Iivari & Lyytinen 1998)

The C*ooperative Design* (Greenbaum & Kyng 1991) has recently aroused considerable interest especially in the CSCW community (Iivari & Lyytinen 1998). CSCW (Computer Supported Cooperative Working) should be conceived as an endeavor to understand the nature and characteristics of cooperative work with the objective of designing adequate computer-based technologies (Kensing & Blomberg 1998: Schmidt & Bannon 1992).

Participatory Design (PD) traces its roots to Scandinavian trade unions, but its ancestry also includes Action Research and Socio-technical Design. It is an approach to the assessment, design, and development of technological and organizational systems that places a premium on the active involvement of workplace practitioners (usually potential or current users of the system) in design and decision-making processes. MUST (Kensing et al. 1998) is a practical method for PD, focusing on cooperation between users, managers, and internal IT professionals who are responsible for the design and implementation of a computer-based system. Computer Professionals for Social Responsibility (CPSR) has sponsored biennial PD conferences since 1990.

Another interesting piece of work was conducted in Denmark. The *professional work practice approach*, also called '*Reflective Systems Development*' approach (Mathiassen 1998) was developed mainly during the 1980s. Mathiassen and his colleagues wanted to investigate how systems development was actually carried out (Lanzara & Mathiassen 1985). They found out that the more experienced the analysts were, the less they followed documented ISD methodologies. Thus, more important for understanding how systems are in fact developed is to analyse and affect actual working practices of the systems professionals in the organization. The PWP approach took much influence from the TU strategy, DUE and MARS projects. Its strength is its emphasis on studying the actual work practices of the systems professionals before attempting to improve them. The empirical base of the PWP approach seems to be stronger than its theoretical underpinnings, which are weakly documented and rather eclectic (Iivari & Lyytinen 1998). That is why this approach is more interesting in the sense of research methodology than the theoretical research approach. I will come back to this approach later on in this chapter when discussing related research in ISD.

The *Activity Theory* approach to systems development is based on the Cultural-Historical Activity Theory developed in Soviet psychology in the 1920's and 1930's (Vygotsky 1978, Leontjev 1978). It has been further elaborated by Engeström (1987a) and applied for example in work development research. According to Iivari and Lyytinen (1998), AT is adaptive in IS research in the sense that both the work practice to be supported and the systems development process can be conceived as distinct but interrelated activities. A weakness of the AT approach and also a strength to some extent is its generality. It is quite a new approach in ISD research and thus it is quite weakly elaborated. Historically, the AT approach is most closely associated with the Trade Unionist approach. Recently, AT has been applied and also elaborated somewhat in information systems research, especially in Finland (e.g. Kuutti 1994, 1996, Korpela 1994, 1999, Leppänen 2000), but also by other IS researchers (e.g. Bødker 1987, Bertelsen 2000). I will concentrate more on the Activity Theory in Chapter 3, since it has been used as a research approach in this study.

The European information systems research has had a tendency to emphasise relevance instead of rigour. During the time that the Scandinavian researchers were concerned with empowering unions, the British researchers had been developing socio-technical design principles (Mumford 1995, 2000) and management philosophies. *Socio-technical theory* emphasises that human needs must not be forgotten when technical systems are introduced (Mumford 2000). It was just these projects which were at the same time being criticized by Marxist theorists in Scandinavia for promoting values that were fundamentally capitalist - increasing productivity and decreasing worker resistance. While these criticisms carry some force with respect to how the principles were often applied within the workplace, they still depended on a mechanistic conception of the labour process organization which pits capitalist demands against humanist concerns and leaves no real space for compromise. (Asaro 2000)

The point of this study. The amount of paradigms, approaches, and perspectives within IS is quite confusing and difficult to scope for a PhD student, but at the same time challenging and interesting. I have purposely examined the different approaches quite broadly in order to clarify how researchers have started to consider people, their work, and the context as important, not only the technical issues. In addition, I wanted to identify the rich context of ISs, in practice and in research. I can only agree with Jones (1997) that seeing IS in the context of social and organizational changes, is both a danger and an opportunity for IS researchers. If we want to understand the role of IS, we need to have a broad understanding of the social processes, and quite a few of us involved in the field are experts in that subject. There is therefore a risk of poor quality research. The opportunity for IS researchers is where the role of the technological element in the social changes is poorly understood, even by otherwise subtle social theorists. It is this conjunction of the social and the technical where IS researchers can contribute which makes IS such a potentially demanding and important field. (Jones 1997)

All in all, I do not see the diversity of IS field as a weakness. I do not believe that IS is 'fading away'. The described paradigms are to help to understand the diversity of the field and different approaches and perspectives, which also have much in common in many places. The diversity is a challenging feature that attracts intelligent people who want to address various problems that interest them. In addition, a diverse field fosters creativity. In fields with unified paradigms, shifts in thinking are rarely challenged. Finally, diversity advances the valued principle of academic freedom. Still, it doesn't mean that 'anything goes', concerning the choices of theoretical frameworks and research methods. (cf. Robey 1996)

The interest of this examination of IS approaches is in ISD research methodologies, not practical ISD methodologies, even if part of the results of my study is to be contributed to the adopted methodology in Africa. I will summarise later in this thesis what we consider important for such a methodology. I think that it is worth mentioning a few words about practical methodologies in general, since the division of methodologies has influenced our research interests in some sense. Also, we can divide ISD methodologies roughly into the *hard* and *soft* methodologies (Korpela et al. 2000b). Hard methodologies are more formal, more concerned with the technological and engineering aspects, and often sold as a commercial package complete with training material and computer assisted tools. Examples of such methodologies are SSADM, ISAC, and Information Engineering. Today various object-oriented analysis and design (OAD) methodologies should be added to this group (Korpela et al. 2000b). However, the recent research literature on ISD methodologies has concentrated less on the technical design of systems and more on the social nature of ISs and their development. Here we refer to soft methodologies (c.f. Hirschheim & Klein 1992). In practice, it is widely believed that systems will improve if systems analysts better understand the users (compare interpretivism and positivism in research approaches). Examples of such methodologies that improve participation in systems development and also support the goals of emancipatory discourse (discourse taking place in an environment free from distortions; Hirschheim & Klein 1992) are Checkland's (1981; Checkland & Scholes 1990) Soft System Methodology (SSM) for managing problem situations; Mumford's (1983) ETHICS (Effective Technical and Human Implementation of Computer Systems) which is a design methodology based on the socio-technical theory of work design; MUST (Kensing & Blomberg 1998) for participatory design; and UTOPIA (Ehn 1989) - project, even if the prototype developed never became a commercial product. Emancipatory methodology refers to better allow users and developers to build ISs which liberates its users from unwarranted social constraints and psychological compulsions, makes the user equal. These soft methodologies have influenced our interest and goal for a better adapted ISD methodology in Africa.

However, the rare empirical studies of ISD methodologies in practice indicate that the use of methodologies is limited or they are not literally applied (see Iivari et al. 2001). This indicates the lack of practicality, which is important to consider concerning the research questions of INDEHELA-Methods project. In that sense, the most interesting approaches seem to be the socio-technical approach, or rather *Participatory Design* approach emphasising user participation, as one inspiration for applied 'Made-in-Nigeria' ISD methodology; the *Reflective Systems Development* as an example of research process to develop methodology in co-operation with practitioners; and *Activity Theory* as a theoretical approach for studying ISD as a work activity in practice.

2.1.4 Information Systems Development – examples of related research

Like in IS research in general, diversity and pluralism is a common feature in ISD research. When somebody wants to find an appropriate framework or theory for studying ISD, the choices are several. I agree with Walsham (1997, p.478) that there is not, and never will be, a best theory, *the theory*. "Theory is our chronically inadequate attempt to come to terms with the infinite complexity of the real world. Our quest should be for improved theory, not best theory, and for theory that is relevant to the issues of our time." The premise in this chapter, when examining related research in ISD, is in theories and approaches used when one wants to study ISD as a pheonomenon, as well as in the contributions of these studies. The examination is not complete, I just want to point to some examples of possible approaches.

Largely referred paradigms for ISD are by Hirschheim and Klein (1989). These paradigms are based on epistemological assumptions (the way in which system developers acquire knowledge needed to design the system) and ontological assumptions (how systems developers view of the social and technical world). They adapted Burrel and Morgan (1979) subjectivist-objectivist and order-conflict dimensions in order to conclude four paradigms: functionalism (objective-order); social relativism (subjective-order); radical structuralism (objective-conflict); and neohumanism (subjective-conflict) (FIGURE 9).

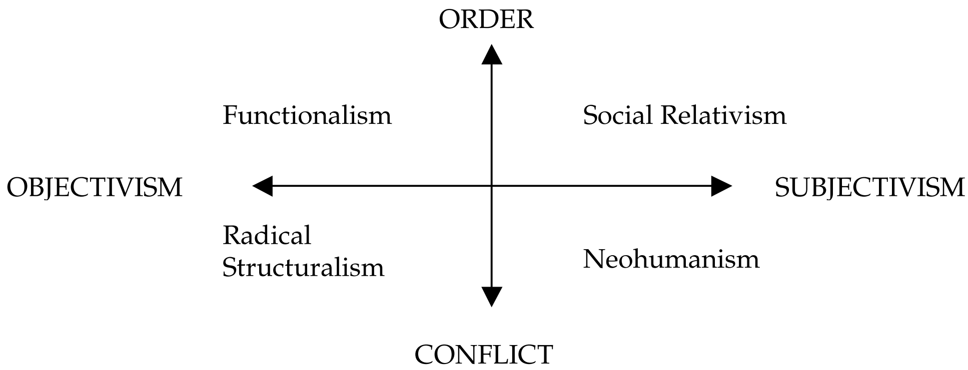


FIGURE 9 Information Systems Development Paradigms (Hirschheim & Klein 1989, p. 1202)

Objectivism refers to applying models and methods derived from the natural science. *Subjectivism* in contrast denies the use of natural science methods for studying the social world. *Order* view emphasises a social world characterized by order, stability, integration, consensus, and functional coordination. *Conflict* view stresses change, conflict, disintegration, and coercion. The *functionalism paradigm* is concerned with providing explanations of the status quo, social order, social integration, consensus, need satisfaction, and rational choice. It seeks to explain how the individual elements of a social system interact to form an integrated whole. The *social relativism* paradigm seeks explanation within the realm of individual consciousness and subjectivity, and within the frame of reference of the social actor as opposed to the observer of the action. The *radical structuralism* paradigm emphasized the need to overthrow or transcend the limitations placed on existing social and organizational arrangements. It focuses primarily on the structure and analysis of economic power relationships. The *neohumanism* paradigm seeks radical change, emancipation, and potentiality, and stresses the role that different social and organizational forces play in understanding change. (Hirschheim & Klein 1989)

TABLE 3 summarises the characteristics of these four paradigms. In practice, information systems development approaches are influenced by more than one paradigm, but the influence from one paradigm is typically dominant (Hirschheim & Klein 1989). Practitioners may or may not be conscious of the philosophical assumptions belonging to alternative paradigms. However, a better understanding of the conceptual foundations of their beliefs including the recognition of other belief alternatives can lead developers to seek creative solutions using the strengths of each paradigm. According to Hirschheim and Klein (1989), researchers should further develop and refine these paradigms and identify involving methodologies.

Usually the key concern in ISD studies has been a development of methodologies. Baskerville and Stage (2001) summarise that there are two contributions concerning that, one to express research results and experiences in the form of an ISD methodology, and another to see methodologies as generalized descriptions of the way in which work is conducted in development organization. They usually assume that methodologies and practices are closely related, the assumption that is less commonly justified. Thus they are referring to the lack of empirical work on ISD and ISD methodologies.

TABLE 3 Summary of the Four Paradigms (Hirschheim & Klein 1989, p. 1210)

Paradigm	Developer archetype	Systems development proceeds	Elements used in defining IS	Examples
Functionalism	Expert or Platonic Philosopher, King	From without, by application of formal concepts through planned intervention with rationalistic tools and methods	People, hardware, software, rules (organizational procedures) as physical or formal, objective entities	Structured analysis, information engineering
Social Relativism	Catalyst or Facilitator	From within, by improving subjective understanding and cultural sensitivity through adapting to internal forces or evolutionary social change	Subjectivity of meanings, symbolic structures affecting evolution of sense, making and shaping of meanings, metaphors	Ethnographic approaches, FLORENCE project
Radical Structuralism	Warrior for Social Progress or Partisan	From without, by raising ideological conscience and consciousness through organized political action and adaptation of tools and methods to different social class interests	People, hardware, software, rules (organizational procedures) as physical or formal, objective entities put in the service of economic class interest	Trade-union led approaches, UTOPIA and DEMOS projects
Neohumanism	Emancipator or Social Therapist	From within, by improving human understanding and the rationality of human action through emancipation of suppressed interest and liberation from unwarranted natural and social constraints	People, hardware, software, rules (organizational procedures) as physical or formal objective entities for the technical knowledge interest; subjectivity of meanings and intersubjectivity of language use in other knowledge interests	Critical-social theory, SAMPO project

In the following, I will give some examples of ISD research approaches which other researchers have applied in order to study the systems development process, namely the ethnographic approach, Reflective Systems Development, accommodating work practices, and the Social Action Theory. I also refer shortly to the Action-network theory, which I consider one possible choice concerning ISD studies. I will also clarify the contributions of the studies and relate the contributions to my research questions.

Ethnographic approach. Ethnographic approach is an example where context and culture are considered crucial. The assumptions often expressed are that cultures have hidden logics which can be uncovered by an intelligent observer who immerses oneself into the membership of that culture (Harvey 1997). In ethnographic research, besides the data sources like interviews and documents, the data is supplemented by data collected during participant observation on the part of the research over an extended period of time. The weakness of this approach seems to be in its feasibility. It is not always possible to participate in the life of the studied domain for such an intensive way and for a long time. In addition, it does not give a practical framework for ISD analysis, even if its' philosophy, concerning culture, is useful. In that sense, it is a very popular research methodology among anthropologists, although it is also a useful approach for IS research.

A method for participatory design, called MUST (Kensing & Blomberg 1998) has been developed by having been inspired by ethnographic approaches and the Scandinavian participatory design (PD) approaches throughout ten projects in Denmark and American organizations. MUST is a method that focuses on early activities in a development process, like most PD methods are. It is based on six principles, namely 1) participation, 2) close links to project management, 3) design as a communication process, 4) combining ethnography and intervention, 5) co-development of IT, work organization and users' qualifications, and 6) sustainability. It is also constituted by five main activities: 1) project establishment, 2) strategic analysis, 3) in-depth analysis of selected work domains, 4) developing visions of the overall change, and 5) anchoring the visions (Kensing et al. 1998). We find many similarities between these issues and the issues we consider important in such a method, like sustainability, participation, and co-development of IT (or IS) and work practices. However, this method does not explicitly include risk management as a part of activities.

Reflective Systems Development. *Reflective Systems Development*, developed by Mathiassen (1998) and his colleagues in Denmark, is a methodology for research and practice of information systems development. The history of this work is based on the 1970s, when influential students at the University of Aarhus organized courses on computers and society, having Kristen Nygaard as a visiting professor in 1974. During the courses they tried to learn about systems development by using and developing different perspectives to understand and criticize methods and practices. They noticed that methods are quite different from practice. During that time, with the help of Nygaard, they started a number of research initiatives in collaboration with Danish trade unions. The most influential project was DUE project (Kyng & Mathiassen 1982). The projects were based on action research in close collaboration with local unions in specific organization. That became part of one of the dominant traditions in early Scandinavian information systems research. They also took some initiatives to formulate concepts and frameworks for systems development research. These initiatives took them beyond the traditional views of computer science to include social perspectives and issues related to organizational change. (Mathiassen 1998)

In 1980 one group of researchers involved in the DUE project initiated a new generation of trade union projects focusing on development of computer-based tools to support and enhance the quality of working life (Bjerknes & Bratteteig 1995): the UTOPIA project (Bødker et al. 1987, Ehn 1988, Kyng 1996), and the FLORENCE project (Bjerknes & Bratteteig 1987). This research used real-world laboratory to explore and develop new professional tools and techniques. The other group continued the action research approach with practitioners. Their goal was to understand, support, and improve the professional practices of systems developers in real-life organizations. This is when the MARS project was formed. The researchers' most important insights were (Mathiassen 1998, p. 77): 1) Systems development methods were seldom, or only partially, followed by experienced practitioners. 2) Each environment had distinct characteristics that played an important role for the success and failure of projects. 3) The most urgent areas for improving practices were project management and more supportive organizational environments. 4) It was possible, but extremely difficult to change working practices. The researchers had changed the perspective from users to professional practitioners. The action research efforts resulted in a book on development processes, project management, and change of work practices (Andersen et al. 1990). In the 1990s the next step was to focus on the reflections involved in designing and engineering computer-based ISs and on the reflections involved in designing and managing development processes. The research continued with the collaboration of a number of international research groups, e.g. Checkland's group. Thus Reflective Systems Development has emerged as an approach which is based on the inspiration from a number of sources. (Mathiassen 1998)

As mentioned already, Reflective Systems Development is methodology both for research and practice. For research purposes, the methodology uses action research as the basic practice form, which is supplemented with experiments and practice studies. The starting points are problems, challenges, and opportunities involved in systems development practice. The action research is based on Checkland's notion of the experience-action cycle (Checkland & Scholes 1990). Research activities yield experience-based knowledge that leads to new practices. The research goals and activities of the methodology are illustrated in FIGURE 10.

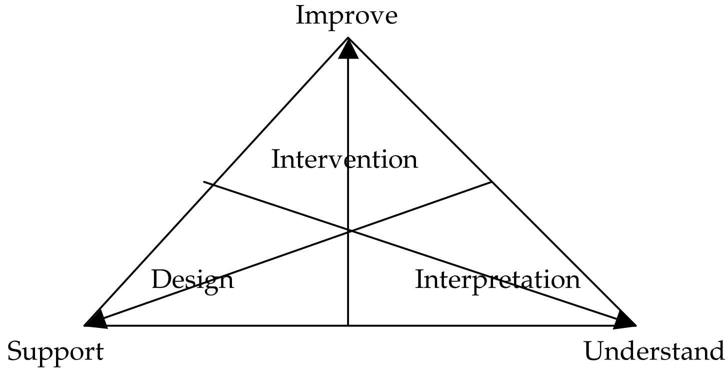


FIGURE 10 Research goals and activities involved in Reflective Systems Development (Mathiassen 1998, p.20: adapted from Vidgen et al. 1997)

According to FIGURE 10 the understanding is based on interpretations of practice. To support practice these interpretations are simplified and generalized and engaged in the design of normative propositions or artefacts. Finally practices are changed and improved through different forms of social and technical intervention. Overall, systems development is intrinsically related to change. According to Mathiassen (1998), every research approach to systems development needs to include this change aspect of practice into its intellectual framework. In Reflective Systems Development it has been done through the use of dialectics (Israel 1979), which helps to understand computer systems and systems development in a social context. The dialectics as an intellectual framework focuses on contradictions, the relations between them, and the opposites and relations that constitute them. (Mathiassen 1998)

In practical situations, we spontaneously know what we are doing, knowing-in-action. Sometimes we have to think what we are doing, we reflect on our action. Professionals usually deal with problematic situations involving complexity, uncertainty, instability, uniqueness, and value-conflict by *reflection-in-action* (Mathiassen 1998: Schön 1983). In systems development the relation between reflection and action depends on whether the view of the process is mainly construction, evolution, or intervention. In construction the issue is to develop a technical solution in response to a given data-processing problem, reflection comes mainly before the action. In an evolution perspective, the problem is not given and stable; reflection-in-action is needed to frame the situation and set the problem. The learning process is iterative. In an intervention, the challenge is to change an organization through development of new or modification of an existing information system. Reflection-in-action is needed to support problem framing and implementation. The process is organized as interventions into a series of problematic situations. System developers should mix these perspectives depending on the characteristics of their efforts, which change during the process. Systems development is a highly collaborative activity which takes place in complex, dynamic arrangements with many actors and organizations involved. Reflection-in-action focuses on individuals, but another perspective, *communities-of-practice*, focuses on groups and organizations. It means that practitioners collaborate and share work experiences through extensive use of narration about issues and problems involved in doing the job. The communities are fluid rather than bounded, they evolve rather than be designed, and they typically cross the formal boundaries of an organization. These two perspectives are claimed to be useful frameworks in systems development, since systems development is a change process that aims at understanding, supporting, and eventually improving practices of other professionals, meaning users of would-be information systems. (Mathiassen 1998)

Hirschheim et al. (1995) suggest some weaknesses concerning the methodology, like lack of clarity and structure, lack of attention given to the cost and effectiveness of the approach, simplistic view of what it takes to improve practices, limited understanding of the role that bias and prejudice play in human understanding, lack of an explicit notion of what it means to change professional practices for the better, and a lack of explicit, theoretically well-founded principles of participation and project organization. Mathiassen (1998) admits these weaknesses and he points to a number of possible avenues for further development of the approach. Referring to Hirschheim and Klein (1989) paradigms (FIGURE 9), Mathiassen puts Reflective Systems Development mainly to the social relativist approach, and to the neohumanism approach, but also to the functionalism approach.

The methodology is very practice oriented and the developers of the methodology see practitioners as the primary users of the research. Its theoretical background is quite pragmatic, and if we consider it for research purposes, it serves more like research methodology, instead of a theoretical background. Also the idea of work development is what we consider important in systems development. Later Mathiassen (2000) applied and developed the methodology further for example in one SPI (software process improvement) project. However, I do not see much connection to my research interests of emphasising risk management and sustainability analysis during the software development project within this methodology, even if the reflection-in-action gives inspiring thoughts.

Framework for accommodating work practices. Baskerville and Stage (2001) are interested in how methodologies are used in ISD work practice. Their study is based on the result that practitioners do not adopt methodologies, they adapt fragments of methods in their work. The practitioners seem to invent and adapt fragments of methods in unique ways instead of following a published methodology. They demonstrate a socio-organizational process for selecting methods in order to design emergent work practices. Work practices are constituted by organization, management, strategy, collaboration, techniques, tools, and evaluation (c.f. MARS project). The framework for accommodating work practices consists of three components: 1. method fragments originated from published methods as well as method fragments that have been innovated through previous practice. 2. work practice helping to define the fragments that might be relevant, and 3. a sociologic process for the on-going accommodation, meaning selection, invention, and combination of method fragments. Baskerville and Stage apply general ethnography to describe the fragment selection process in a case of IT development in a bank. They describe different encounters between system developers and their community. Thus their interest and framework focus on ISD methods, including work practices as one component of the framework. Still, the elements of work practices mostly include the same basic elements than for example the activity theory, just having different concepts for them.

Social Action Theory. Hirschheim et al. (1996) use the *social action theory* perspective when viewing ISD as one example of human activity that deals with design. According to them, ISD also deals with behavioural issues concerned with the changing conditions and forms of social behaviour brought about by the design outcomes. In addition, all design activity takes place in a historical context under specific social influences and it must be drawn upon by a set of practices that are shared by a community. They ask for theories that offer conceptual frameworks that clarify conditions, means, contents, constraints, and objectives of socially organized human behaviour (Habermas 1984). Social action theories focus on the question why and how all knowledge, including our knowledge of ISD, is socially contingent. Accordingly, Hirschheim et al. (1996) apply one of the most prominent modern social action theories – *Critical Social Theory* – to the problem of understanding ISD. The basis of the Critical Social Theory is in 1930s Germany, at the Institute of Social Research, and in writings by Habermas (1984, 1987). Habermas sees four orientations in changes caused by ISD; instrumental orientation, strategic orientation, communicative orientation, and discursive orientation. The first two orientations emphasize control. The instrumental orientation treats people as mere physical objects whereas the strategic orientation as intelligent agents. Both orientations share, however, the idea of control in the sense of affecting objects (human and non-human) in an environment in such a way as to achieve given ends. This requires that an agent predicts the consequences of his or her behaviour in the environment and based on this prediction chooses an optimal action. In the communicative orientation the emphasis is on creating shared meanings (and mutual understanding) through sense-making. The primary emphasis in the discursive orientation is on argumentation which ensues when claims made during communication are called into question or for clarification. Actors can change their orientation very quickly in ISD and all these action types are present in ISD (see Ngwenyama 1991).

Habermas' social action theory provides a basis to organize and classify social processes that make up ISD. However, Habermas is too general in his approach to state what is being changed in ISD, the type of 'substance' being changed during ISD. Hirschheim et al. (1996) extend Habermas' theory by adopting some concepts from Amitai Etzioni's *The Active society* (1968). Based on his concepts, Hirschheim et al. grouped classes of ISD changes to the domain of change. By change they mean 'purposive' change through human action. They argue that any change, be it ISD, technological change, societal change or so, is based an the actor's underlying set of beliefs and assumptions about the nature of the domain he or she is changing. According to them, the principal domains of change in information systems development are technology, organization, and language. Firstly, all information systems contain some technological components. Secondly, language is fundamentally the most easy to change (codes, symbols, meanings) because of its conventional rule-based character, and finally, the organization as an object of change includes people and collectives. Organizations are structured and patterned based on deeply rooted, sedimented, and often unconscious practices and beliefs. Based on the Habermas' orientations and Etzioni's domains, they form the federated framework with object system classes (Lyytinen 1987, Welke & Konsynski 1982). The matrix of the object system classes is exhibited in TABLE 4.

TABLE 4 Object system classes and examples of possible objects (Hirschheim et al. 1996, p. 17)

Domains	Orientations			
	Control		Sense-making	Argumentation
	Instrumental	Strategic	Communicative	Discursive
Technology	*Information Technology Systems* Hardware and telecommunication configuration; Program structure and modules; Database and file structures			
Language	*Formalized Symbol Manipulation Systems* Data models and dictionaries; Data integrity mechanisms; Screen and form designs; Model management systems.	*Manipulative Communication Systems* Definition of terms and rules; Communication channels; Access rights; Data integrity	*Symbolic Interaction Systems* Speech acts; Intentions; Meanings; Metaphors	*Systems for Rational Argumentation* Arguments; Warrants; Breakdowns; Pragmatic inference
Organization	*Mechanistic Social Systems* Tasks; Decision Processes; Business processes; Organizational structures	*Political Systems* Power Structures; Resource dependencies; Interest groups; Sources of authority; Indirect influence; Negotiated orders	*Cultural Social Systems* Values, beliefs; Myths, rituals; Negotiated meanings and practices	*Systems for Institutional Checks and Balances* Domination free discourse; Justification and minimization of power; Truth and justice; Due process

The matrix with nine object system classes identifies the major intellectual structures that explore possible changes brought about in ISD (Hirschheim et al. 1996). For example the matrix suggests how the objects in the domain are to be approached by a systems developer. Accordingly, the matrix helps to cope with the complexity and ambiguity of ISD. It forms a relatively stable and interrelated set of possible abstractions that can be applied to frame and understand ISD situations.

The domains and orientations are helpful to clarify the 'what' aspects in ISD, but the research and practice of ISD is as much concerned with 'how'. Besides object system classes Hirschheim et al. (1996) develop a generic model to describe the ISD process (FIGURE 11).

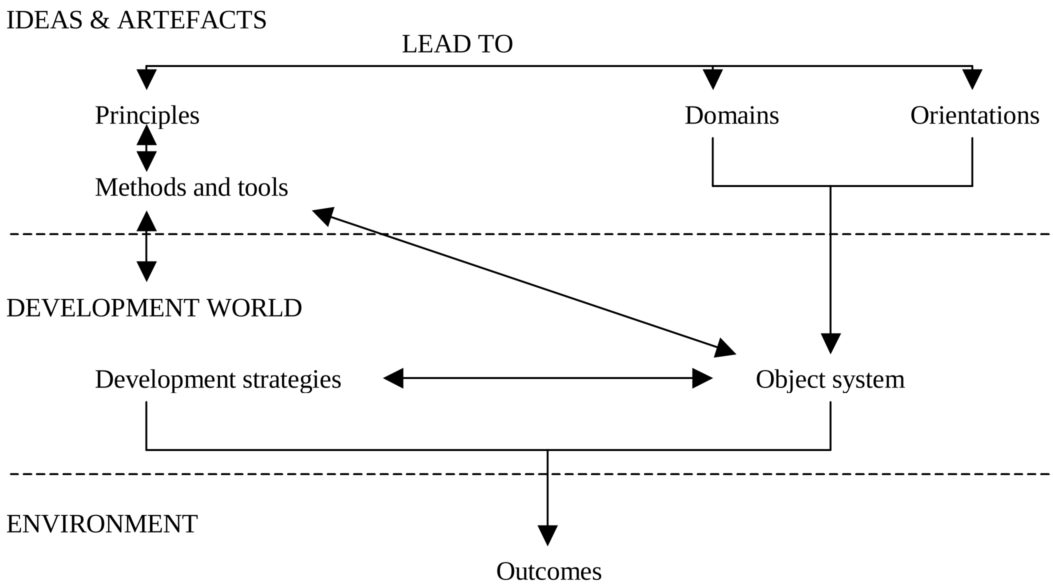


FIGURE 11 Generic model of information systems development (Hirschheim et al. 1996, p. 23)

FIGURE 11 is divided into three levels: ideas and artefacts, development world, and environment. Ideas and artefacts build up the stock of knowledge and material artefacts required to carry out a development process. It consists of components like domains, orientations, principles, and methods and tools. The development work consists of concrete processes, people, and actions that carry out development activities. It consists of development strategies and corresponding object system classes. The environment consists of changes brought about by ISD. It covers the outcomes or consequences, both intended and unintended. For each element the matrix of object system classes can be applied by looking for specific principles, methods, strategies, and outcomes for each class.

The matrix of object systems seems to be very comprehensive in perspective of the information system. The rows and columns in the matrix reflect the establishment of independent research approaches and directions in the 'fragmented adhocracy' (Banville & Landry 1989), but their interrelations clarifies how they at the same time are dependent upon one another (Hirschheim et al. 1996: Etzioni 1968). By combining the matrix with the generic model of ISD, the framework spells out some of the major conceptual elements that constitute targets of ISD studies, such as methods and tools and their use in varied settings. Viewing ISD through the generic model offers a means to locate various method and tool uses, process and impacts and to suggest new methods, tools, and principles for less developed research targets. (Hirschheim et al. 1996)

If we want to consider ISD as a comprehensive phenomenon or as a work activity, the framework is still quite restricted, since the perspective is solely on information systems, the object system. For example, the object system class '*cultural social systems*' abstracts from ISs their ability to affect patterns and processes of interaction which facilitate sense-making. It means that the organization is seen as consisting of loosely structured and overlapping roles with social interaction patterns continuously changing and evolving. Thus IS – besides hardware, software, and data – includes procedures, skills requirements, value laden social objects, habits, belief systems, and the like. Therefore, IS cannot be studied without understanding a complex technical and social infrastructure surrounding it and a history of social and local practices that depend on and shape its use. The benefits of an IS are contingent upon a host of social conditions. There exits a duality between the organic and mechanistic social systems. Very true, but based on the model, this is not considered if we are interested in changing the performance level. Secondly, the matrix basically wants to illustrate the affects of IS to domain objects. The matrix is very helpful and inspiring to understand ISD and IS as a research field, but it is not meant as a framework if we want to study ISD as a work activity. On the other hand, the generic model is useful to illustrate the process of ISD. The model includes the important elements involved in systems development, only the collective and contextual aspects are missing.

Ngwenyama (1991) summarises the fundamental assumptions for critical social theory: 1) People have the power to change their world. 2) Knowledge of the social world is value laden. 3) Reason and critique are inseparable. 4) Theory and practice must be interconnected. 5) Reason and critique must be reflexive in practice. Critical social theory does not have its own research methodology, but many interpretative methodologies can be adapted to its needs. The theory aims to integrate the fundamental knowledge interest, which is technical, practical, and emancipatory, into a holistic approach to inquiry and intervention.

Actor-network theory. The basis of the *actor-network theory* is in sociology of science and was pioneered by Michel Callon (1986) and Bruno Latour (1987). Later work has also included a focus on information technology (e.g. Nandhakumar & Vidgen 2001: Monteiro & Hanseth 1996) and the theory can be considered to be a development of one strand of the wider school of the social construction of technology (Bijker et al. 1987). Contrary to the socio-technical theory, where the social and technical elements are considered as separate, the actor-network theory treats them as inseparable, and indeed argues that people and artefacts should be analysed with the same conceptual apparatus. It is a framework for analysing the interaction between an action and the influencing factors in a dynamic setting (Baruah 2000). The action-network theory examines the motivations and actions of groups of actors who form elements, linked by associations, of heterogeneous networks of aligned interests (Walsham 1997). When introducing IT into an organization, it is a dialectical process where human and non-human actors interact with each other in the process of designing and implementation of an IS (Baruah 2000). When applied in a particular context, the theory attempts to explain the processes where relatively stable networks of aligned interests are created and maintained, or alternatively to examine why such networks fail to establish themselves (Walsham 1997). As an example of using an actor-network theory in ISD research, Walsham and Sahay (1996) describe an attempt at the creation of a network of aligned interest for the development and use of GIS for district-level administration in India. Baruah (2000) examines a case study of the Tax Administration in India, the process of computerisation the allotment of a unique taxpayer identification number, and the success of it, by using the actor-network theory.

A major strand of criticism of the actor-network theory is that it addresses the local and contingent, but that it pays little attention to a broader social structure. The other broad area of criticism concerns its stance on moral and political issues. Walsham (1997) concludes that no researcher who has used this theory prohibits the use of it in the IS field, but they do suggest some cautions or qualifications. The theory can be complemented by other social theories which take a better account of the broader social structures or moral issues. (Walsham 1997)

The point of this study. When a researcher is selecting an appropriate research approach or methodology for example when wanting to study how ISD is practiced in a given context, the first thing to do is to check how others have done it. The first choice of course is how to approach the problem, is it going to be a positivist paradigm or interpretive. Many times the research consists of both positivist and interpretive methods. Choosing a research method, I believe, is a matter of research object and realistic alternatives. What is perhaps the most problematic part is to choose a proper theory. Of course one can also build a theory by oneself. However, it is argued that theory is an integral part of appropriate methodology, it makes a difference during observation (Weick 1985).

Järvinen (1999) summarises that the theory should include 1) a boundary that describes the domain of interest, 2) key constructs within that domain, 3) the values those constructs can take on, and 4) the relationships among key constructs. Constructs, or concepts, are the items of interest in a domain. In research we use concepts referring to different concrete and abstract objects and entities. We can divide elements into classes or groups by classifying them (Järvinen 1999). According to Markus (1983) theory can be tested in two ways: 1) The basic assumptions underlying the theories can be examined and compared with facts in the 'real world', and 2) the implications for action derived from theories can be tested for their usefulness to implementers.

However, I believe what Weick (1985) writes about theories: they are limited because none can be acceptable in all the demands of generality, accuracy and simplicity. Thus no theory will be fully satisfactory, but it does not mean that theories should be discarded. As McGuire (1984) articulates, all theories are also correct. He argues that empirical confrontation is not to test the correctness of the theory. Contrary, it is a discovery process to clarify what that theory means, disclose its hidden assumptions, and clarify the conditions under which it is true or false. It is to discover where theories make correct predictions.

All in all, what is the contribution I wanted to make by this literature review of related ISD studies? Using whatever theory, it is important to know and understand the other related studies in order to position one's study within the field. When you understand other perspectives it is easier to see 'the conditions under which the theory is true or false' (see above). Without reflecting your study with others, it is impossible to elaborate on theories or construct new theories, or even understand your own results. In the following, I will continue my literature review by discussing successful ISD and further, successful IS use.

2.2 From successful ISD to successful IS use

After we have studied how information systems development is practiced in a given context, the next question should be how this systems development can be practiced so that it will be beneficial and useful for developers (provider) as well as for users (client), not to mention other stakeholders like management, owners, maintainers, final customers (the clients of client), and so for. The question is how to make everyone a winner (c.f. Boehm 1989).

The number of failures in information system development (ISD) projects is high in spite of development in information technology (IT) and the increase in the knowledge and the experience of the users (see e.g. Lyytinen & Hirshheim 1987, Conrow et al. 1997). A survey by the Standish Group (according to Conrow et al. 1997) of 365 respondents and 8380 commercial software-intensive projects indicated that 53 % of the projects were 'challenged': they were over budget, behind schedule, or had fewer features and functions than originally specified, and 31 % of the projects were cancelled. On average, at the time they were completed or cancelled, these projects had cost increases of 189 % and a schedule slippage of 222 % versus the original estimations. In addition, the completed projects had an average of only 61 % of the originally specified features and functions

The issue of IS failure has been studied and published in IS literature for a long, e.g. Brooks (1974), Lucas (1975), Keen (1981), Lyytinen and Hirschheim (1987), Lyytinen (1988), Beynon-Davies (1995), Lyytinen and Robey (1999). Failures in information systems development usually refer to cost overruns or project delays, unmet user needs or even cancelled projects (Barki et al. 1993, Cule et al. 2000). Failures in IS use usually refer to un-reached efficiency and effectiveness of an activity to be improved, inconsistent use or under utilization of the system, non-achievable system objectives or even abandonment of the system. In most serious failures the consequences can lead to serious accidents, like the loss of the space shuttle Challenger (e.g. Wise & Debons 1986, Perrow 1984). This kind of accident can be caused by hardware component failures, software failures, and/or human factors (Bilinski 1986), but many times these accidents are difficult to foresee, because it involves failures on several parts of a system which are linked in complex and subtle ways (c.f. Beynon-Davies 1995). Perrow (1984) illustrates these kinds of accidents and considers them as 'normal accidents', rather than exceptional events. One severally quoted example of information system failure is the case of the London Ambulance Service Computer Aided Despatch (LASCAD) project, where probably 20-30 people have died because of the system breakdown and the late arrival of ambulances because of that (Beynon-Davies 1995).

Heeks et al. (1999, p.2) identify four main forms of failures when they studied some health informatics systems. The main four forms of failures are:

1. The total failure of a system never implemented or in which a new system is implemented but immediately abandoned.
2. The partial failure of an initiative in which major goals are unattained or in which there are significant undesirable outcomes.
3. The sustainability failure of an initiative that succeeds initially but then fails after a year or so.
4. The replication failure of an initiative that succeeds in its pilot location but cannot be repeated elsewhere.

Even if the system itself is working as planned, the risk of failure in IS use exists if the change process in the user organization is not managed and controlled properly. Lyytinen and Hirschheim (1987, Lyytinen 1988) define IS failure as an *expectation failure*. It means a gap between stakeholders' expectations expressed in some ideal or standard and the actual performance. The concept is pluralistic, since it assumes that failures depend on the plural values of stakeholders. It is also political since most failures are resolved through political means such as compromises, bargaining, truce, and bribery.

Lucas writes in 1975 that because of the concern over technology people seem to have ignored the fact that almost all information systems exist within the context of an organization. Thus, the most fundamental purpose of ISD is to facilitate user organization or further, some work activity in user organization by means of some software application. The reason why users need that software, or rather a new form of IS, is that they too provide some product or service to their clients, and the IS is to improve the production of these services (Korpela et al 2001b). In that sense, the activity of ISD is part of a service chain, it is not a separate activity.

Lucas (1975) gives some reasons, relevant to today as well, why ISs frequently fail. *First*, since multiple parties – management, users, developers – are involved in the design and operation of IS, all these groups must work together to develop and operate successful systems. *Second*, a number of factors are involved in the design and operation of successful systems, like technical, behavioural, situational, and personal factors. I would add organizational and contextual factors as well. The complex relationships among these factors must be all considered. If any variable is ignored, the system are likely to fail.

In general, the fact that ISD usually causes changes in user organization is the starting point for a successful process. Alter (1999) provides some typical success factors for example for work systems and information systems in general (TABLE 5). Work system means a system where participants perform a business process using different resources to produce products in an organization.

Heeks et al. (1999) emphasise that when seeking to understand failure, it must be realized that there will be no single blueprint for success or failure in organisational change (Poulymenakou & Holmes 1996). Instead, it must be recognised that there are situation-specific factors for each information system implementation, which will determine success and failure and, hence, strategies for success. A successful IS will be one that tends to match its environment in relation to technical, social and organisational factors; these latter include the perceptions of key stakeholders.

TABLE 5 Success factors for work systems and information systems (Alter 1999, p. 35-36)

Success factors for work system in general	
Context	Consistency with culture
	Cooperative decisions about work methods
	Low level of turmoil and distraction
Infrastructure	Adequate technical infrastructure for the work system
	Adequate human infrastructure for the work system
Customer and product	Product design consistent with customer needs
	Adequate product performance
Business process	Fit of business process with other elements
	Adequate resources for business process
	Effective operational management
Participants	Appropriate skills and understanding
	Interest in doing this type of work
	Motivation to do theis work in this setting
	Ability to work together to resolve conflicts
Information	Adequate information quality
	Adequate information accessibility
	Adequate information presentation
	Adequate information security
Technology	Ease of use
	Adequate technology performance
	Maintainability
	Compatibility with technology in related systems
Success factors for information systems in general	
Context	Extensive experience with information systems
	Positive beliefs about information systems
Infrastructure	Adequate support by the IT staff
	Adequate training on content
	Adequate training on plumbing
Customer and product	Product design consistent with customer needs
	Adequate product performance
Business process	Comfortable fit with the work system being served
Participants	Familiarity with using computers
	Confidence computers will not be used for de-skilling, job elimination, etc.
Information	Information appropriately tailored to the situation
Technology	Adequate internal design
	Adequate compatibility with technology in related systems
	Adequate documentation
	Adequate performance of technology

However, there is a major problem here: if the IS were to exactly match its environment, it would not change that environment in any way. Yet the purpose of ISs is to support and improve the functioning of organizations. There must therefore be some degree of change. On the other hand, if IS tries to change too much this brings with it a risk of failure. Success becomes more likely when change is limited. Overall, there is a trade-off between change and risk for IS. (Heeks et al. 1999)

In relation to what has been said so far about the success or failure of the ISD process, my thesis concerns both software development and IS use. The purpose is not to conduct a failure analysis for some cases, but first of all to clarify how systems are developed in practice, and then focus on problems in IS development and use. Lyytinen (1988) divides IS failures into two phases: development failures (goals, technology, economy, view of organization, process characteristics, self-image) and use failures (technical solution, data problems, conceptual problems, people reactions, complexity). Accordingly my interest is in constraints in systems development, and more specifically *risk factors* of software development, as well as risk factors of IS use, for which we have chosen a concept of *sustainability* as a metaphor for appropriate, useful and sustained IS solutions in user organization. These two aspects should go hand in hand and taken care of by project management in order to confirm a successful systems development process.

2.2.1 The concept of risk

The concept of *risk* has several definitions in information systems. Webster's Dictionary (p. 1660) defines it as "an exposure to the chance of injury or loss, a hazard or dangerous chance". Rowe (1988) defines risk as "the potential for realisation of unwanted, negative consequences of an event". Boehm (1989) defines *risk impact* or risk *exposure* (RE) as

RE = Prob(UO) x Loss(UO)
where Prob(UO) is the probability of an unsatisfactory outcome, and
Loss(UO) is the loss to the parties affected if the outcome is unsatisfactory.

According to Van Scoy (1992, p.3), there are two factors that comprise a risk:

1. Probability or likelihood that it will occur.
2. Loss resulting from its occurrence.

Thus risk is involved with a possibility or probability and negative consequences. *Risk item* or risk *factor* is a factor that may cause a risk situation. It is a characteristic that affects the probability of a negative event occurring. Risk factors can be general or project-specific (Boehm & Ross 1989). *Risk event* is the situation where risk occurs (cf. Charette 1989). Risk outcome is the resulting situation after the risk event. It is usually followed by reaction, which results in some risk effect (Kontio 2001). We are referring to risk consequences.

Accordingly, the concept of risk includes risk factors, probability, risk event, outcome, and consequences (usually negative). The consequences and the value of risk impact are dependent on the stakeholders that are involved. When the risk scenarios are defined, their impact on the project is described through the stated goals by each stakeholder group. *Utility loss* is the harm a stakeholder experiences due to occurred risk (Kontio 2001).

Ropponen (1993) defines risks to be dealt within two worlds: current and future. Risk is something that might occur in the future: a possibility, not a certainty. ISD projects and risky situations represent a current world. The future world (in risk scenario) consists of unwanted negative consequences and potential loss due to realized risk event and project failure. To describe risk we modify the definitions of Ropponen (1993) and Kontio (2001), as illustrated in FIGURE 12.

As a result, we define software project risk in the terms of Barki et al. (1993) as:

> *a product of uncertainty associated with project risk factors and the magnitude of potential loss due to project failure.*

Risk is a fuzzy concept referring to different situations. It is essential to distinguish between risk, problem, and constraint. Risk is something one cannot predict (for sure) and which can negatively impact on a project. On the other hand, there should be a possibility to affect the recognized risk (Ropponen 1993). Usually the level of control over the situation varies. Risk is a concept to see future problems. Risk exists so far as it does not occur, after that it will be a problem (c.f. Fairley 1994). Problem is something that exists in a project and which can be harmful and taken care of. Constraint is something in the environment, also known but not usually able to control. Both a problem and a constraint can turn into a risk if the situation in these elements changes.

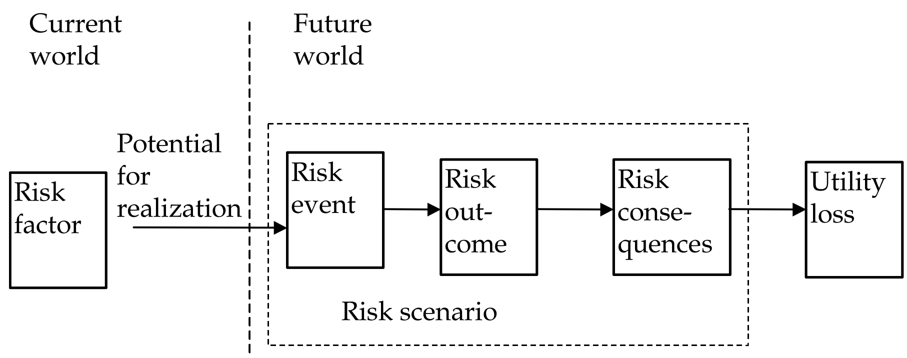


FIGURE 12 Description of risk

Usually risks are considered as a negative issue. In FIGURE 12, the risk management is directed at risk factors. It tries to influence the future world through the management of risk factors. If risk management is a success, the non-realization of potential risk will turn to sustained utility in the future world. Thus we can divide the above figure into two potentials; utility loss and utility gain (or sustained utility), as described in FIGURE 13.

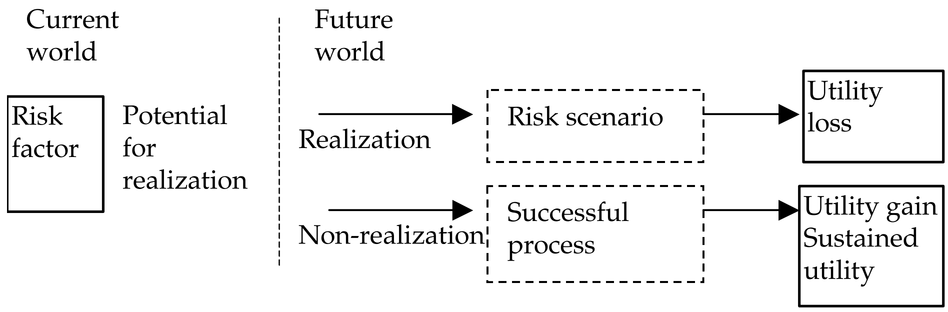


FIGURE 13 Description of risk potentials

Next I will give some examples of how risks in software development and IS use have been described in the IS literature, and also what has been said about the risk management during the ISD process.

2.2.2 Risk factors of software development

The first stage in managing failures in software development projects is to identify the risk factors. Several articles have been published concerning risks associated with software projects (e.g. Alter & Ginzberg 1978, McFarlan 1981, Curtis et al. 1988, Boehm 1989, Genuchten 1991, Barki et al. 1993, Willcocks & Margetts 1994, Charette et al. 1997, Conrow et al. 1997, Känsälä 1997, Moynihan 1997, Keil et al. 1998). As a result there have been various lists of risk factors with some similarities and some differences. However, as Alter and Ginzberg (1978) already notice, the main point is not to provide an exhaustive list of risk factors, but to keep in mind that one should assess the risk factors present in any system development situation. The knowledge of how identified risk factors change over time is valuable in order to develop suitable risk management methods. Risk factors change over time because of the development of technology and organizations. That is why researchers should from time to time conduct rigorous risk studies where they ask (Schmidt et al. 2001): 1.) What are the typical risk factors software project managers face? 2.) Which of those factors do software project managers consider more deserving of their attention? and 3.) which countermeasures are most effective in mitigating risk, given a particular set of risk factors?

Giving some examples of collected risk factors over time, I will present the ones by Alter and Ginzberg (1978), Boehm's (1989, 1991) 'top ten' list, which has been among the best known and used lists of common risk factors, and Keil et al. (1998, see also Schmidt et al. 2001) cross-cultural list of risk factors. Our work will give an additional contribution to the study of Keil et al. (1998) by giving new perspective of the third world. This contribution is presented in the result part of this thesis.

Alter and Ginzberg (1978) were interested in uncertainties in MIS implementation. They suggest that the likelihood of successful MIS implementation can be increased by identifying the key uncertainties at each stage of the development process and devising strategies for the range of possible results. They conclude that the message based on the interpretation of their results is that planning in the early stages of the implementation process may well be the real key to project success. They identify eight deviations which were particularly clear-cut causes of the difficulties in MIS implementation in their study (Alter & Ginzberg 1978, p. 26: Alter 1975):

1. Nonexistent or unwilling users;
2. Multiple users and designers;
3. Disappearing users, designers, or maintainers;
4. Inability to specify the purpose or usage pattern in advance;
5. Lack of loss of support;
6. Lack of prior experience with similar systems;
7. Inability to predict and cushion the impact on all parties;
8. Technical problems and cost-effectiveness issues.

McFarlan (1981) suggested and operationalized three dimensions influencing the risk inherent in a project; project size, experience with the technology, and project structure.

Boehm's (1989, 1991) "top ten" list indicates the primary sources of software project risks:

1. Personnel shortfalls;
2. Unrealistic schedules and budgets;
3. Developing the wrong functions and properties;
4. Developing the wrong user interface;
5. Gold-plating;
6. Continuing stream of requirements changes;
7. Shortfalls in externally furnished components;
8. Shortfalls in externally performed tasks;
9. Real-time performance shortfalls;
10. Straining computer-science capabilities.

Boehm and Ross (1989, p. 908) emphasise that there are two primary classes of project risks: 1) Generic risks, which are common to all projects, and which are covered by standard development planning techniques, and 2) project-specific risks, which reflect a particular aspect of a given project, and which are addressed by project-specific risk management plans. According to them, the most common project-specific risks are personnel shortfalls, unrealistic schedules and budgets, inappropriate requirements, shortfalls in external components and tasks, and technology shortfalls or unknowns.

Many of these earlier risk studies are however quite old and based on anecdotal evidence or they are limited to a narrow portion of the development process. There has been a lack of systematic attempts to identify software project risks by inquiring the opinions of those who actually have experience in managing such projects, as was pointed out by Keil et al. (1998). They conducted an international Delphi study in three different socio-economic contexts (Hong Kong, USA, Finland), where experienced software project managers identified and ranked the most important risks. One aim of the study was to obtain some insights into the cultural variation in risk item perception and ranking. The list of up-to-date risk factors and the consciousness of possible environmental biases were considered as valuable results, for researchers as well as for project managers. Keil et al. ended up at 53 different risk factors. They organized the list into a set of 14 groups based on the source of the risk (see more detailed in Schmidt et al. 2001):

1. Corporate environment;
2. Sponsorship /Ownership;
3. Relationship Management;
4. Project Management;
5. Scope;
6. Requirements;
7. Funding;
8. Scheduling;
9. Development Process;
10. Personnel;
11. Staffing;
12. Technology;
13. External Dependencies;
14. Planning.

In addition to the complete list of risk factors, Keil et al. (1998, p.78) identified a common set of eleven risk factors that the three independent panels, representing very different countries and cultures, selected as the most important items. They suggest that these eleven risk items are, in some sense, universal:

1. Lack of top management commitment to the project;
2. Failure to gain user commitment;
3. Misunderstanding the requirements;
4. Lack of adequate user involvement;
5. Failure to manage end user expectations;
6. Changing scope/objections;
7. Lack of required knowledge/skills in the project personnel;
8. Lack frozen requirements;
9. Introduction of new technology;
10. Insufficient / inappropriate staffing;
11. Conflict between user departments.

Even if the study by Keil et al. (1998) is cross-cultural and thus valuable in order to understand the cultural dimensions of risk perception, their samples are all from technologically developed countries. The developing countries perspective is still missing, thus leaving the study a little biased. I am going to come back to the study by Keil et al. (1998) later on in this thesis, since we repeated the Delphi study in Nigeria in order to get a developing country's perspective into the matter. When introducing research methodology, I will present the used Delphi method in more detail. Also, when presenting the results of our risk study, I will refer to the results by Keil et al. (1998) by comparing our risk list to theirs.

An interesting outcome of Keil et al. (1998) study is a risk categorizing framework. They discovered that risks thought to be the most important are often not under the project manager's direct control. So the notion of control and its relationship with perceived importance enabled them to develop a framework for mapping different types of risks. The framework is presented in FIGURE 14.

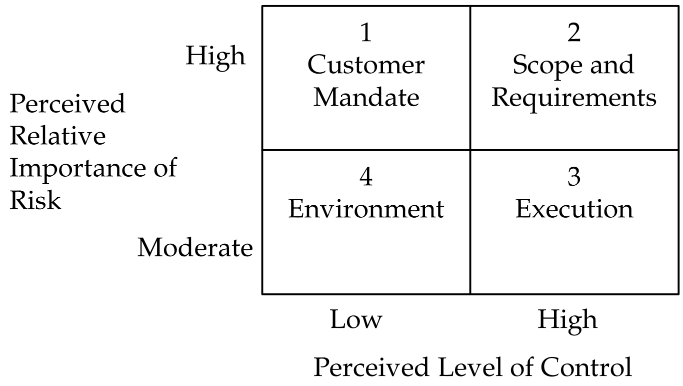


FIGURE 14 A risk categorization framework (Keil et al. 1998, p. 80)

In the framework, the dimension of a perceived relative importance of risk means the relative importance of a particular risk factor in relation to other risk factors. The dimension of a perceived level of control represents the degree into which the project managers perceived that their actions could prevent the risk from occurring. (Keil et al. 1998)

The first quadrant, customer mandate, indicates that successful projects have the commitment of both senior management and those who will actually use the system. Example risks of this quadrant are a lack of top management commitment and failure to gain user commitment. These are quite critical items over which the project manager had comparatively little control. Risk mitigation strategies should create and maintain good relationships with customers and promote customer commitment to the project. Keil et al. (1998) suggest approaches like 'Theory-W' by Boehm and Ross (1989), which involves structuring the project to meet the 'win' conditions of various stakeholders. Approaches should also focus on managing the end-user expectations. Thus project managers must have relationship management, trust-building, and political skills in order to manage risks of this quadrant. (Keil et al. 1998)

The second quadrant, scope and requirements, refers to the ambiguities and uncertainties that arise in establishing the project's scope and requirements. Examples of those kinds of risks are misunderstanding the requirements and not managing change properly. These are critical items, but ones that a project manager should be able to control. Risk mitigation strategies should emphasize the management of ambiguity and change. The scope and requirements should become clearer and expectations more realistic as time progresses. The project manager should be able to draw a line between desirable and necessary functionality of the system. Keil et al. suggest techniques like multicriteria decision-making and function-point analysis. (Keil et al. 1998)

Quadrant three, execution, concerns the risks involved with actual execution of the project. Examples of risks are associated with poor project management, like inappropriate or insufficient staffing, lack of effective development process methodology, poor estimation, and improper definition of roles and responsibilities. These risk items are considered moderate and controllable by a project manager. The risk mitigation strategies should emphasize internal evaluations with external reviews to keep a project on track.

Thus project managers must follow an established development methodology and proactively anticipate and respond to events that can threaten the development process. (Keil et al. 1998)

The last quadrant, environment, refers to the environment that exists both inside and outside the organization. Examples of risks here include changing scope and objectives, and conflicts between user departments. This category also includes external risks like natural disasters or changes in the competitive environment. These kind of risks are those over which the project manager has little or no control. They are not considered important, but when they become acute, they can be dangerous for a project. Keil et al. suggest techniques like contingency planning and disaster planning. (Keil et al. 1998)

The advantage of the framework by Keil et al. (1998) is that instead of focusing on the individual risk, it is a higher-level framework for considering different types of risks and strategies for addressing each type. Keil et al. also compared their risk list to some earlier lists. I will come back to that comparison when I discuss the results of our risk study.

2.2.3 Software development risk management

Software development risk management is not an easy task. Still, many software project disasters could have been avoided or strongly reduced if there had been an explicit identification of high-risk elements at an early stage of the project (Boehm 1991). Kontio (2001) refers to some industrial reports (still rare) on software risk management and summarises that even if the reports cannot provide quantifiable data about the benefits of risk management methods, they do provide indications that some benefits exist. All in all, the importance of risk management has been agreed upon and reported in IS literature (e.g. Alter & Ginzberg 1978, Charette 1989, Boehm 1989, Boehm & Ross 1989, Boehm 1991, Lyytinen et al. 1996, Ropponen & Lyytinen 1997, Lyytinen et al. 1998). McFarlan (1981) suggests a portfolio approach to IS management referring to three serious deficiencies in IS management: the failure to assess individual project risk, the failure to consider the aggregate risk of the portfolio of projects, and the lack of recognition that different projects requires different managerial approaches.

Risk management is not a one-time activity, but an ongoing process of identification, assessment, and control of risks, integrated into IS management (c.f. Smith et al. 2001). Boehm (1989) suggests a simple *Risk Management Plan* as a minimum task to integrate into everyday project management: 1) Identify the project’s top 10 risk items (the number 10 is not a requirement), 2) present a plan for resolving each risk item, 3) update the list of top risk items, plan, and results monthly, 4) highlight risk-item status in monthly project reviews, compare with previous month’s rankings, status, and 5) initiate appropriate corrective actions.

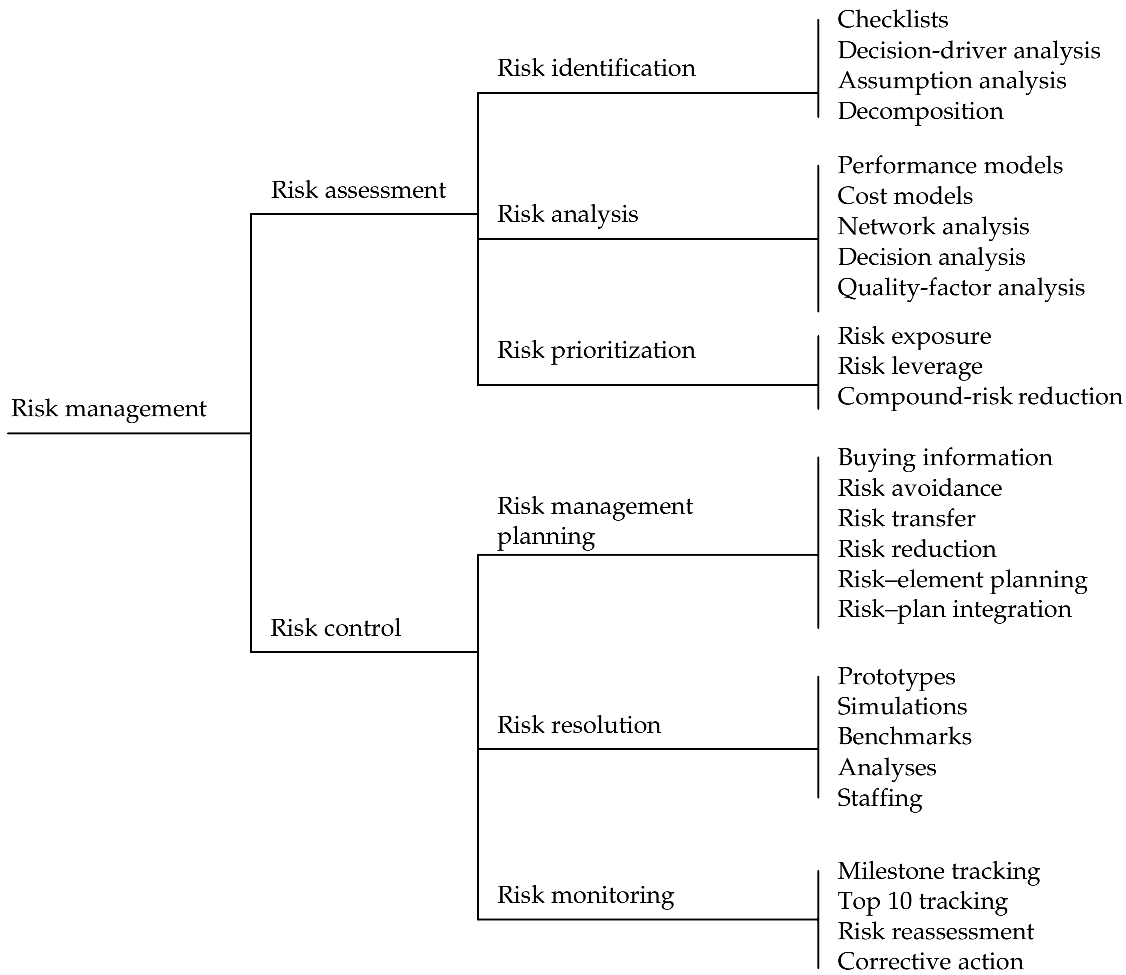


FIGURE 15 Software Risk Management Steps (Boehm 1991, p. 34)

As a more comprehensive managerial method, Boehm (1991) presents software risks management steps, where the practice of risk management involves two primary steps, risk assessment and risk control (FIGURE 15). Further, risk assessment includes risk identification, risk analysis, and risk prioritisation. Each of these steps include suitable techniques appropriate to manage each step. Risk control involves risk management planning, risk resolution, and risk monitoring, with suitable techniques. (Boehm 1991)

Many other risk management methods include the same kind of steps when compared to Boehm's steps above. For example, based on Kontio's (2001) review, SEI's Software Risk Evaluation method (Sisti & Joseph 1994, Monarch et al. 1996) is structured with the following tasks: identify, analyze, plan, track, and control, all related to communication. A central element of SEI is the risk taxonomy (consisting of the structure of product engineering, development environment, and program constraints) and associated questionnaire that project manager should go through. Kontio's (2001) own contribution is the Riskit Method, consisting of a process including risk management mandate definition, goal review, risk identification, risk analysis, risk control planning, risk control, and risk monitoring. The method extends the traditional risk management methods by including the goals and stakeholders as essential entities for risk definition. A stakeholders perspective and failure analysis has been emphasized earlier for example by Lyytinen and Hirschheim (1987), and stakeholder goals and risk management by Boehm and Ross (1989) when they introduced the win-win approach. The Riskit Method defines goals and expectations to be dependent on the stakeholders involved. It views the evaluation of risks as a multiple criteria decision making problem and uses the appropriate techniques to evaluate all affected goals when comparing loss scenarios (compared to risk description above). The Riskit Method includes several artefacts or tools for conducting each step and also software application called eRiskit (Kontio 2001).

In order to implement risk management practices and principles, project managers must insert these into the existing life-cycle management practices. Boehm (1989, 1991) presents a risk-driven software process model called *spiral model*, where risk considerations determine the overall sequence of life-cycle activities. The spiral model is quite comprehensive and complicated, so an easier way to start is the above mentioned Risk Management Plan by Boehm (1989). It is easy to use and provides early improvements. At the same time project managers can become familiar with other risk-management principles and practices.

Organizational theories have provided a useful lens to examine organizational changes involved in ISD, when studying and developing risk management approaches and methods (Alter & Ginzberg 1978, Keen 1981, Lyytinen et al. 1996, 1998, Ropponen 1999). Lyytinen et al. (1996, 1998) and Ropponen (1999) use a socio-technical model of organizational change (Leavitt 1964) to analyse the content of risk management approaches. The socio-technical model (see also Mumford 1983, Keen 1981) views organizations as multivariate systems consisting of four interacting components – task, structure, actor, and technology (FIGURE 16).

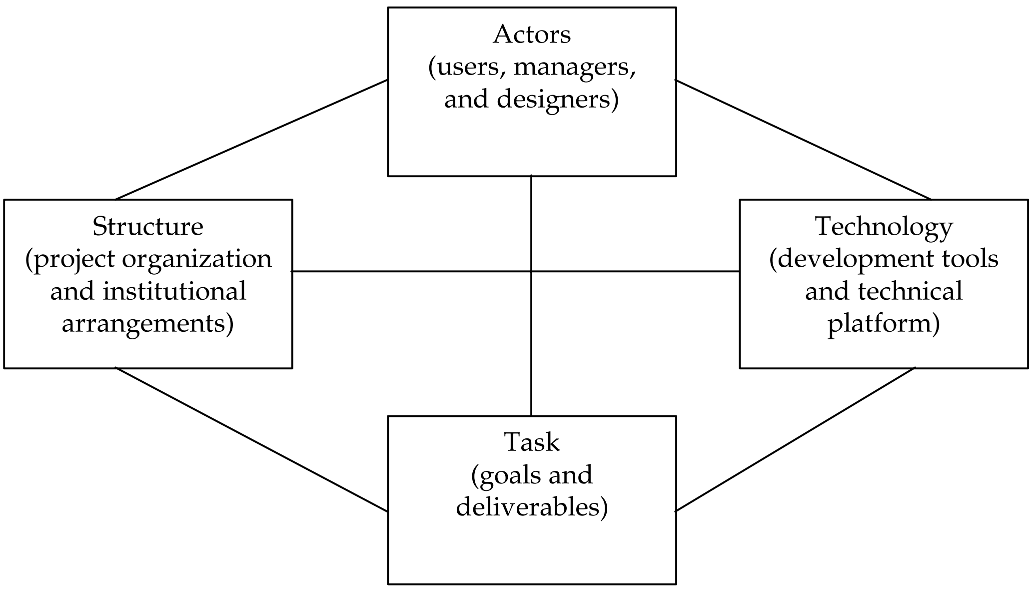


FIGURE 16 A Socio-technical Model of System Development (Lyytinen et al. 1998, p. 241: Leavitt 1964)

The elements in FIGURE 16 are strongly related, and changes in one component will affect others. The model also postulates that if one component's state is incongruent with others, it will disturb the whole system. Accordingly, the goal of the management system is to control and maintain the system in balance, otherwise, in an extreme situation, this incongruent can lead to failure in systems development. Lyytinen et al. (1998) provide this socio-technical model of risk management to direct managerial attention to largely ignored risk areas and to orchestrate action. The framework can be applied in three levels or environment: management environment, project environment, and system environment (Lyytinen et al. 1996). The framework can also be used as a research framework when studying software risk management behaviours in practice (c.f. Ropponen 1999).

Referring to FIGURE 14 of risk categorization framework by Keil et al. (1998), the authors continued to classify risks and their management by creating a new framework, '*a risk classification and behaviour model*' (Cule et al. 2000). In the framework each category requires a different managerial behaviour and risk mitigation strategies (FIGURE 17). The first criteria for categorization was how controllable risk items are for a project manager; some of the risk items are totally under control, while others were clearly outside the manager's direct control. They identified these two patterns as *Inside and Outside risks*, from the project manager's point of view. The inside risks further include two subgroups, *task* and *self*. The task group refers to risks reflecting an active control aspect of the project management. The other subgroup, self, relates to the project manager's own capabilities and understanding. The outside risks are divided into *client* and *environment*. The risks concerning clients are risks that the project manager could not control but perhaps influence. The environment risks could be neither controlled nor influenced by a project manager. (Cule et al. 2000)

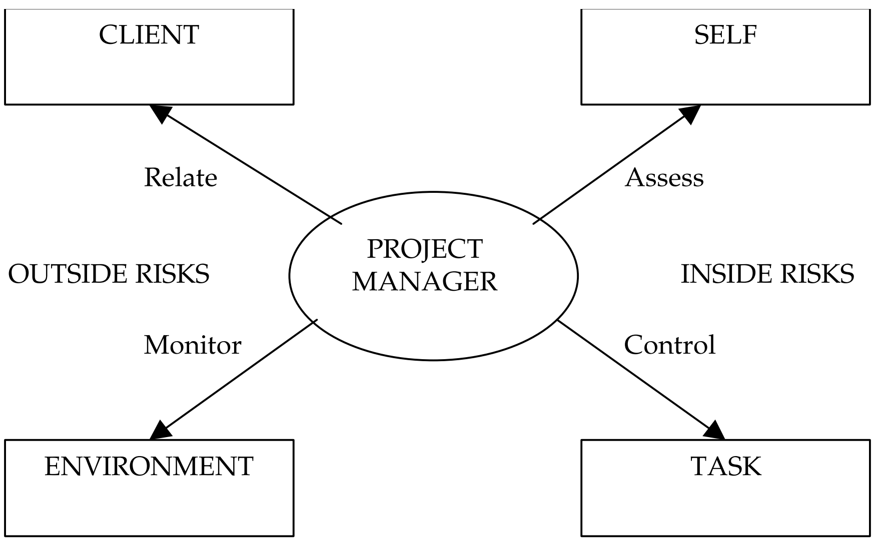


FIGURE 17 A Risk Categorization and Behavior Model (Cule et al. 2000, p. 67)

Self Assessment: The risks in Self category appertain to the project manager's abilities, capabilities, and knowledge when managing software projects. So a project manager needs to assess his/her capabilities against the projects needs. Assessment may need to be done by others (other or upper managers or independent auditors). A project manager can also benchmark other projects or other organizations to compare and learn. There are also assessment mechanisms, for example Process Maturity Model (Watts 1995). (Cule et al. 2000)

Task Control: The project manager should take care of the task, it is a subject of project management texts and education. Any experienced project manager should be able to control the task. (Cule et al. 2000)

Environmental Monitoring: These risks are those about which the project manager can do little: 'Act of God.' The other type is rooted in the industry environment, competition, and government action. These types of risks require that the project manager be cognizant of them even if he/she can do nothing to prevent their occurrence. The monitoring responsibility requires the project manager to be knowledgeable about what is going on in the environment, and how it might affect the project. (Cule et al. 2000)

Client Relationships: These project risks are associated with those people internal to the company with whom the project manager deals. These are the people who are essential to the successful outcome of the project. It is essential that the project manager establish relationships with these individuals and groups. (Cule et al. 2000)

According the Cule et al. (2000) this model provides for the IS project manager a new way of viewing the responsibilities he or she has, but not in isolation. The responsibility for a successful project is not only the project manager's. The model lays open the need for active participation and proactive responsiveness from the group of stakeholders.

The point of this study. Project managers rarely apply specific risk management methods (Ropponen 1999), mainly due to low understanding and knowledge of software risks or their management. Such methods should be therefore easy to use and flexible to apply to different situations. Even if the defined steps of risk management (e.g. Kontio 2001, Boehm 1991) seem to be practical, they demand changes in management activity, which is not easy for a practitioner (e.g. project manager) to do without help, and more importantly, without the commitment of top management and the whole organization.

In addition, for being too heavy or cumbersome, many methods are limited focusing only on inside elements of the project, leaving environment and stakeholders outside of the analysis. The methods provided by Kontio (2001) and Cule et al. (2000) are well-come exceptions emphasising these contextual factors as important as well. There might be other goals for a software development project than efficiency, schedule, and cost, or product quality. There might be different requirements for example for sustainability of the would-be application.

The main issue in my mind is that before any risk management activity can be taken, the risk itself must be identified. As presented earlier, many authors have proposed a list of risk factors to be used as a check-list to manage different risks (e.g. Boehm 1989). Even the most common risk factors vary over time and organization, and projects can also have some unique risks. Thus such check-lists are biased and limited in many cases to be relied on solely, though they can give an excellent starting point for risk identification.

All in all, integrating risk management into the IS project management is an essential part for a successful IS project. We can also take a broader perspective. Risk management can be a powerful approach to dealing with the complexities and uncertainties that surround technological change and its management in user organization. Conventionally, risks are defined narrowly with software engineering, but today IS is becoming more and more of an integral part of a company's existence, systems are becoming more interconnected and thus executives are beginning to realize that serious human and organizational risks are associated with the use of IS. Therefore, effective risk management is an important issue to both IS and business managers (Smith et al. 2001). However, we must be realistic, since no approach provides any 'silver bullet' *per se* for ISD success (c.f. Brooks 1987).

2.2.4 Risks in the use of information systems

Information systems are supposed to be beneficial for user organizations, and many times it is the situation (cf. Dhillon & Backhouse 1996). There are also risks associated with the use of information technology in organizations. The studies of risks in IS use can be classified into two categories, according to Dhillon and Backhouse (1996). The first category includes studies of security issues of information systems, dealing with computer viruses, hacking, systems failures, and access control. This group largely consists of computer scientists who are concerned with providing complex technological solutions (see e.g. Baskerville 1991). The other category includes studies by accountants and business managers who are concerned with information systems in terms of environmental and organizational contexts, gaining to cost-justify solutions. They are increasingly considering human and organizational aspects as well (e.g. Willcocks & Margetts 1994). The concept of success or failure is thus *conditional*, according to different stakeholders (Mäkeläinen et al. 1996). However, to include organizational implementation and operation and maintenance in a project is a surprisingly common issue in the IS field (Alter 2001b).

Mäkeläinen et al. (1996) approach success or failure in 'use' situation with an ONION model, where they evaluate the state of deployment of an IS, and the problems, in four levels: individual, group, organizational unit, and enterprise. They connect the analysis to a work activity, and the objectives of the activity are to a great extent determined by the next larger context. The four levels of analysis are consistent with our ideas (see chapter 3 and ActAD). However, the model is still quite general and it does not provide tools to analyse the activity in a rigorous way.

Smith et al. (2001) organized a focus group (Judd et al. 1991) with 17 senior

IS managers to discuss risk management in IS projects. They discussed for example the sources of risks and resulted in the following sources which I consider to be realized in the use of the would-be information system, but which must be already considered during the ISD project:

- *Financial risk*, meaning that the financial return of an IS project should be greater than the amount invested in it.
- *Technology risk*, including risks with new and untried technology, as well as technology performance, scalability, reliability, and stability. It also includes operating risks (e.g. technology failure can prevent business to be conducted) and strategic risks (wrong choices or poor implementation can be a competitive threat).
- *Security risk*, referring to the reliability of the electronic environment (e.g. application security, data integrity, and network defensibility).
- *Information risk*, including privacy, decision-making risk, and strategy development risks.
- *People risk*, because people are a source of uncertainty. They respond subjectively to change, so their reactions are difficult to predict. People risk also includes poor management, and things like pressure, burnout, and loss of face.
- *Business process risk*, since IS usually make changes in business processes thus creating a risky situation. When processes change, information flows change and this often creates operational havoc (Simons 1999). Also lack of technological usability, poor help desk and support, inadequate training, and unanticipated results contribute to business process risks.
- *Management risk*, since every project has a set of vulnerabilities and dependencies which need to be managed. These include for example schedule, budget, functionality, compatibility, relationships, expectations, and communication.
- *External risk*, caused by the growth of IS outsourcing, IT subcontracting, ERP (Enterprise Resource Planning) systems, and other pre-packaged software.
- *Risk of success*, since projects can be as unprepared for success as they are for failure. For example, the volume of transactions can be higher than expected or the users see more potential in an application than was anticipated. These issues can lead to demand for expansion of a project, for which the structures of a company are inadequate.

In addition to the above mentioned risks, there could be *internal abuse* (malicious or felonious destruction, theft, abuse etc. by company insiders), *competitive risk* (negative reactions by customers, competitors, suppliers, etc. to the company's IT initiatives), and *reputational risk* (negative reactions by the public at large, the media, the government, etc. to a company's IT initiatives) (Markus 2000). Also, the source of problems could come from the natural environment (e.g. lightning, earthquakes, floods, even electromagnetic and other interference including cosmic radiation and sunspot activity), or from animals (e.g. sharks, squirrels, monkeys, birds), and infrastructural factors (poor electricity, loss of air conditioning) (Neumann 1995).

The risks in the use of IS should already be considered in early design activities of the ISD process, but it must be noted that risks can change during the process. They are not static (Markus 2000). Most often these considerations are related to systems security, the need for backup in the event of a systems crash, or the prevention of fraud. As can be seen from the above mentioned risks, besides the technical and formal structures of an organization and the would-be information system, the informal and pragmatic structures should also be included into the sustainability analysis of the information system (cf. Dhillon & Backhouse 1996, Margetts 1994). We have chosen the concept of sustainability to describe the final position of the information system in use. Next I will clarify the concept in more detail.

Sustainability. The term 'sustainable development' was used at the time of the Cocoyoc declaration on environment and development in the early 1970s. Since then it has become the trademark of international organizations dedicated to achieving environmentally benign or beneficial development. The term has served to catalyse debate over the relationship between economic change and the natural-resource base in which it is grounded. The term 'sustainable development' suggests that the lessons of ecology can, and should, be applied to economic processes. (Redclift 1992)

In Webster's Dictionary the term sustainability has not been defined, but the term 'sustain' has been defined as to "to support, hold, or bear up from below" (Webster p. 1917).

Oyomno (1996, p.21) writes that "sustainability comes in to ensure that the improvements in the quality of life that are achieved as a result of developmental efforts are continuously enjoyed over a long period of time and that further improvements are attainable as the environments change". Thus sustainability has a futuristic nature. Oyomno refers to Brundtland Commission (1987) by defining sustainable development as "*meeting the needs of the present without compromising the ability of future generations to meet their own needs*". We can apply the ideas of 'ecodevelopment' by saying that the organizational changes or development is dependent on *basic needs*, *self-reliance*, and *environmental sustainability*, concerning the organization itself and the environment it is acting in. The environment is not just what is outside the organizations. It can be looked upon as more of a process rather than form, as the result of a set of relationships between physical space, natural resources, and a constantly changing pattern of economic forces. The environment in the international economy is an internationalised environment and one which often exists to serve economic and political interests far removed from a specific physical 'location'. The development cannot be separated from the historical processes. There is a history to the process through which the environment has been internationalised. The environments of developing countries bear the imprint of colonial history, but the social formation established under colonial rule varied widely. So the sustainability also has a historical nature. (See Redclift 1992)

It can be said that sustainable development is dependent upon *sustainable technology* in this research context of ours. By sustainable technology we mean that the improvements achieved by technology are sustained and are enjoyed over time, the technology itself is usable and useful, the possible changes in an organization or in environment are intended, manageable, far-reaching and humane, and that further improvements are possible to conduct smoothly. According to Oyomno (1996, p. 22) the sustainability of technology is functionally dependent upon three main variables:

- the level of demand for the technology;
- the appropriateness of the technology to the application environment; and
- the availability of local technological capacity to sustain its beneficial use.

Oyomno (1996) studied governmental organizations, but I believe that his definition is suitable for other organizations as well. We can illustrate sustainable technology by FIGURE 18.

Demand can be seen as a measure of the extent to which the use of a certain technology is required in an organization. A technology that is in less demand is less likely to be sustained. Technology demand was further defined to be functionally dependent upon three variables; (1) The extent to which the applications or activities to which the technology is put, or developed, are critical to the proper functioning of the organization. (2) The expected productivity that accrued to the organization as a result of using the technology. (3) The value of the outputs from the technology to the organization. (Oyomno 1996)

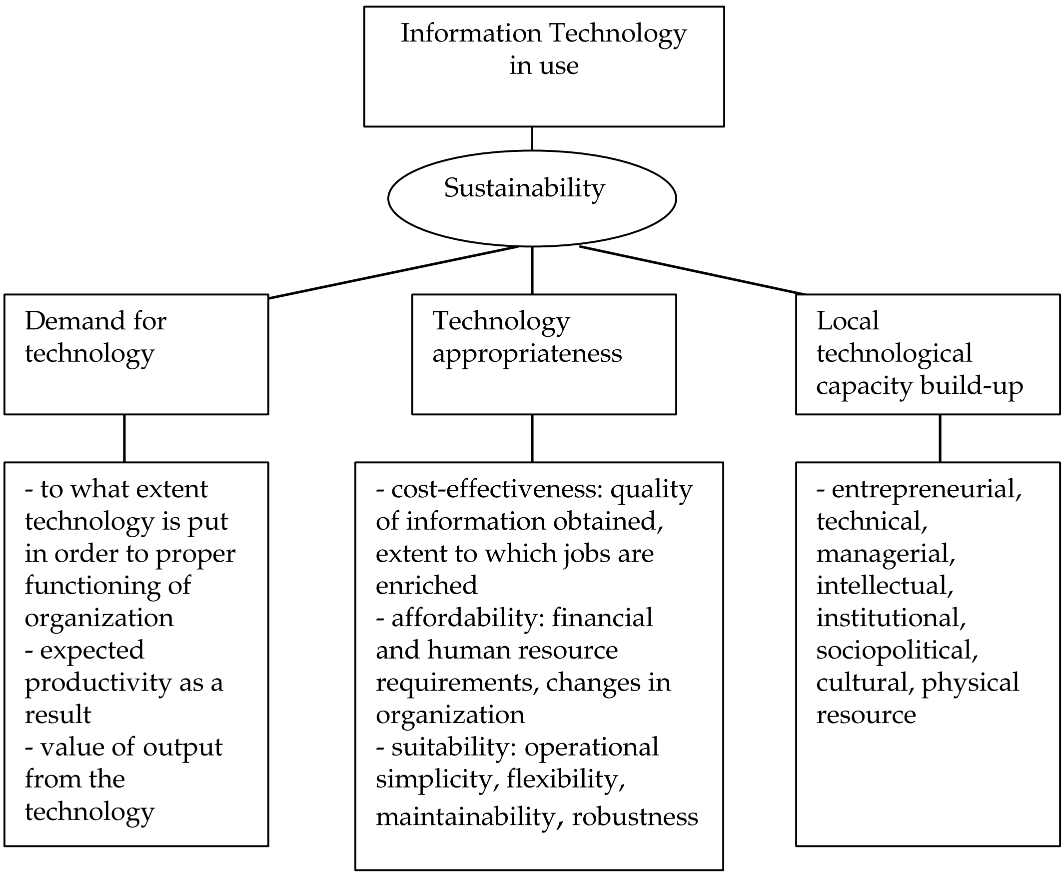


FIGURE 18 Sustainable technology

The appropriateness of technology determines its acceptability and subsequent adoption and institutionalization in a new organisational setting. Technology may be useful and in high demand and yet inappropriate for sustained use. For example, it may be too expensive or sophisticated to be supported and maintained on local organization's resources. Technology appropriateness can be divided into three variables: (1) cost-effectiveness, (2) affordability, and (3) suitability. Cost-effectiveness can be defined in terms of quality of information obtained and the extent to which using technology enriches the jobs of individual members of the organization. Affordability can be studied in terms of financial and human resource requirements of the technology, and in terms of the adaptive changes required to fit technology into organizations. Technology suitability can be studied in terms of its operational simplicity, flexibility, maintainability, and robustness. (Oyomno 1996)

According to Kiggundu (1989) *local technological capacity* is entrepreneurial, technical, managerial, intellectual, institutional, socio-political, cultural, and physical resources and infrastructure that exist in an organization and its immediate environment. It is related to the extent to which an organization is able to utilize effectively its new and existing technology. For continued beneficial use of a new technology, the essential preconditions is the creation of an organizational resource infrastructure. For example, many times new technologies are introduced into a developing country organizations through technical assistance to facilitate their rapid installation and utilization. However, persistent use of technical assistance, even years after the acquisition and installation of the technology, is a strong indication of a lack of adequate local technological capacity build-up for the technology in these organizations. (Oyomno 1996)

Since ISD involves several stakeholders with different goals, and the process is also very much of a work development process in user organization in certain context, besides software development, risks of sustained use of IS should be considered during the process. It is reasonable to extend the traditional software project risk management ideas to cover also the use of IS. There is a lack of such a framework having both these aspects considered, focusing on *the total lifecycle risks* (Markus 2000). Lyytinen and Hirschheim (1987) discuss both development failures and use failures and ask for the integrated framework. I believe that the question is not relevant but only in very complex computer systems, like in nuclear power stations or space shuttles (c.f. Neumann 1995), but also in information systems in organizations where people are dealing with their everyday business.

2.3 Summary

This long chapter aims to provide a theoretical foundation based on my research interest. The concepts of the information system (IS) and information systems development (ISD) are broad and the research approaches vary accordingly. In this literature review I wanted to clarify the development of IS research to include qualitative methods and social aspects, besides the technical aspects. Contextuality and different stakeholders are nowadays emphasised in many approaches, and interpretive paradigm is accepted as a serious research approach. The historical perspective is critical in order to understand the existing situation.

ISD is considered as a change process, also affecting the user organizations and their structures. ISD is a work development effort at the same time, in most of the cases. Thus the failure or success of the ISD process concerns both software development and the use of resulting IS in a user organization. Risk management aims to promote successful ISD projects, so risk management should be extended to also cover successful use of ISs. Especially in developing countries the sustainability of ISs is essential to promote socio-economic and human development. These issues are discussed in more detail in Chapter 4 in this thesis.

3 THEORETICAL FOUNDATION BASED ON ACTIVITY THEORY

This chapter provides basic background information about the activity theory and its origins. The basic elements of the structure of human activity are presented, as well as the idea of Developmental Work Research, both based on Engeström's (1987a) work. The activity theoretical approach is further elaborated by Korpela (1994), resulting in Activity Analysis and Development methodology, which is presented as a research approach to this thesis. The literature review consists of the research related to the information systems and activity theory.

3.1 Why activity theory

One purpose of my research is to study how information systems are developed in Nigeria, what are the practices and constraints in that work. So my approach concerning ISD is quite within work practices and in a socio-economic context which gives environmental factors affecting the work. The activity theory, already strongly applied in the INDEHELA-Methods project beforehand, seems to give the appropriate tools when studying ISD as a phenomenon and as a work practice. Especially the elaborated activity analysis framework, which I will introduce in this chapter seems to be a good practical framework for my study. After all, it has been developed keeping practitioner orientation in mind.

The strengths for the activity theory are based on its long historical roots. It is a philosophical and cross-disciplinary framework for studying different forms of human practices as development processes, with both individual and social levels interlinked at the same time (Kuutti 1996). The strengths are the interaction in a social context and the idea of dynamics and development. According to Iivari and Lyytinen (1998) the strengths of the activity theory are: 1) activity theory can serve as one theory of social action, a promising background theory, 2) the concept of activity is a potentially useful concept to structure the organizational context, and 3) it already has an expanding research community. They also name some weaknesses, like: 1) the theoretical background is very general, 2) it is not fully elaborated as an ISD approach, and 3) the language context is weakly addressed. The generality of the background theory is not always a weakness, if it allows you to apply and elaborate the theory for future research. In this study we have elaborated the theory (mostly based on Finnish elaboration by Engeström 1987a) to be more practical for ISD research. During the elaboration, for example, we noticed that even if the activity theory considers context as important, it is quite restricted concerning the broader socio-economic context of the environment.

All in all, the reasons why I think the activity theory will serve my requirements for a research approach in this study are in brief: it is theoretically strong with a long history, it considers the context, the history of development of an activity and the dynamics of an activity as important, it is both general and detailed focusing on activities and the network of activities (concerning the relations of activities), and it also focuses on contradictions within and between activities. Last but not least, it offers a practical framework for studying ISD work practices.

3.2 Activity Theory

The history of the activity theory has three main origins. One is the 18th and 19th century classical German philosophy from Kant and Hegel, in which the concept of activity was fist introduced, another consists of the writings of Marx and Engels, who also elaborated on the concept of activity further, and the third source is the Soviet cultural-historical psychology, founded by Vygotski (1978), Leontjev (1987) and Luria (1976) (Kuutti 1991, Engeström 1999). After World War II, activity theory has mostly been developed and applied in psychology, learning, education, cognition, and child development. Today it is more and more multidisciplinary. During 1980s and 1990s the activity theory has encompassed research topics like development of work activities, implementation of new cultural tools such as computer technologies, and issues of therapy (Engeström & Miettinen 1999). The First International Congress of Research on the Activity Theory was organized in Berlin in 1986. In 1987, an international scientific society for research based on the activity theory was founded, namely ISCRAT (*International Society for Cultural Research and Activity Theory*). ISCRAT has organized a new international congress every four years.

The basic unit of analysis in the activity theory is an activity, which ties individual actions to context. Leontjev (1978) elaborated on Vygotsky's concept of 'human activity mediated by tools and signs'. He structured a three-level hierarchical structure where collective activity is the holistic basic unit of analysis (TABLE 6).

Activity consists of actions, which in turn consist of operations. Activity is a collective event with shared object or motive. Actions are conscious behaviours with a defined goal, carried out by an individual or a group. An operation is like a non-conscious event within an action. The actions are always connected with a frame of reference created by the corresponding activity. An activity includes different actions, while one and the same action can belong to different activities. Operations are well-defined routines used by the subject subconsciously. All the operations are actions at the beginning, but when the action has been practiced long enough, the action will became an operation. An operation is much more fluent, but at the same time a new action is created with a broader scope, containing the recently formed new operation. The border between action and activity is also blurred, and movements are possible – into both directions. (Engeström 1987a, Kuutti 1991)

TABLE 6 Leontiev's three-level model. (Engeström et al. 1990, p. 140)

Unit	Directing factor	Subject
Activity	Object / motive	Collective
Action	Goal	Individual or group
Operation	Conditions	Non-conscious

As examples of activity – action – operation division can be given a software project and research (Kuutti 1991). 'Completing a software project' is an activity, where we can find actions like 'programming a module' or 'arranging a meeting'. There are also operations like 'using operating systems commands' or 'selecting appropriate programming language constructs'. In research, the activity can be 'carrying out research into a topic'. The actions includes 'searching for references' and 'participating in a conference', as well as 'writing a paper'. On an operational level there are 'using logical syllogisms' and 'selecting appropriate wording'.

I argue that the identification of activity and action, or action and operation, is always case sensitive and interpreter sensitive. The speculation of this level is out of the interest of this thesis. In addition, I am not going to examine throughout the origins and applications of the activity theory. Rather, I am going to start from the direction elaborated by Engeström (1987a) and a clear presentation by Kuutti (1991, 1994), continuing to examine what has been published in terms of the information systems and activity theory.

3.2.1 Core concepts of Activity Theory

The basic element – unit of analysis – of the activity theory is an activity. Activities, in which humans participate, have special properties. Firstly, an activity has an *object* (also called motive) and activities can be distinguished according to their objects. An object can be a material thing, but also less tangible or totally intangible, like a plan or common idea. The activity usually ends resulting in some *outcome*, which is a motive of the existence of an activity (transformation of an object). An activity is also a *collective phenomenon*. It has a *subject*, individual or collective, who understands its motive. A subject and an object are in interaction mediated by some '*tool*' or mean. Thus an activity exists in a *material environment*, including processes, relations, shared concepts, meanings etc, *and transforms* it. An activity is a *historically* developing phenomenon. *Contradictions* are the force behind the development of an activity.

An activity is realized through conscious and purposeful actions by participants. (Engeström 1987a, Kuutti 1991)

Kuutti (1991, 1994) gives a nice, brief, and compact overview of the basic ideas of the activity theory. He highlights three of the key ideas of it (1994, p.52):

> *1. Activities as basic units of analysis.*
> When studying any social phenomenon, the problem is always a dichotomy between individual and social. Actions without context are meaningless. There is a need for an intermediate concept – a minimal meaningful context for individual actions – which must form the basic unit of analysis. This unit is called activity. Because the context is included in the unit of analysis, the object of research is always essentially collective, even if the main interest lies in individual actions.
> *2. History and development*
> The activity theory claims that the differences between groups of actors or work practices cannot be really understood without seriously analysing the historical development which has led to them. The activities, including all the elements within it, are under continuous development which is not linear or straightforward. Only through a historical analysis can the sources for the change or development be understood, and without knowing these sources or reasons to guide the development can be worthless.
> *3. Mediation*
> One basic idea of the activity theory is that the relations between and within the elements of an activity are mediated by various artefacts. These artefacts have been created and transformed by actors during the development of an activity, thus there are always historical remains. The artefacts can be instruments, signs, procedures, machines, methods, laws, forms of work organizations, accepted practices etc. Due to the nature of artefacts, they should be never treated as 'given'. They should be included into the study as integral and inseparable components of human functioning.

Engeström (1990b) would add 4.) *Inner contradictions as the source of change and development*. Activity systems are characterized by inner contradictions, they are not stable and harmonious systems. Development can be understood by tracing disruptions, troubles, and innovations at the level of concrete modes of the activity, both historically and current.

3.2.2 Basic structure of the concept of activity

Engeström aims to illustrate by a simple model the concept of activity. As a starting point he refers to Vygotsky's basic notion of mediation in a simple stimulus-response relationship (FIGURE 19). Accordingly Engeström explains his model with the following four criteria (Engeström 1987a, p.39):

> "*First*, activity must be pictured in its simplest, genetically original structural form, as the smallest unit that still preserves the essential unity and quality behind any complex activity.
> *Second*, activity must be analysable in its dynamics and transformations, in its evolution and historical change. No static or eternal models will do.
> *Third*, activity must be analysable as a contextual or ecological phenomenon. The models will have to concentrate on systemic relations between the individual and the outside world.

Fourth, specifically human activity must be analysable as a culturally mediated phenomenon. No dyadic organism-environment models will suffice."

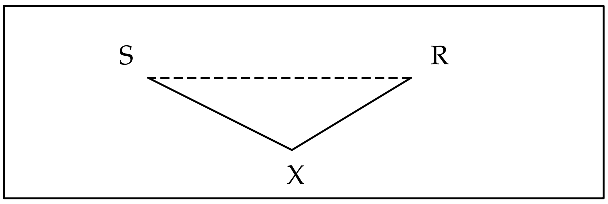


FIGURE 19 The structure of the mediated act (S: stimulus, R: response, X: mediation by tools and signs) (Vygotsky 1978, p. 40)

'Cultural mediation' means that the relationships between the main components of the concept cannot be defined arbitrarily but are situation-bound and have developed historically in the course of the cultural process. (Kuutti 1991)

Engeström (1987a) developed a model – *the structure of human activity* – where the basic triangle of the *subject – object – instrument* has been extended. He completed the model by adding rules, community, and division of labour (FIGURE 20). *Rules* mediate between community and subject, *division of labour* mediates between community and object, and *instruments* mediate between subject and object. Object is transformed into the *outcome*.

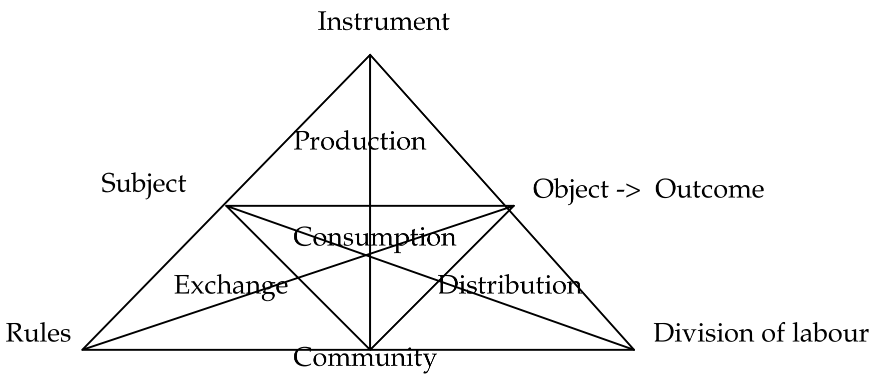


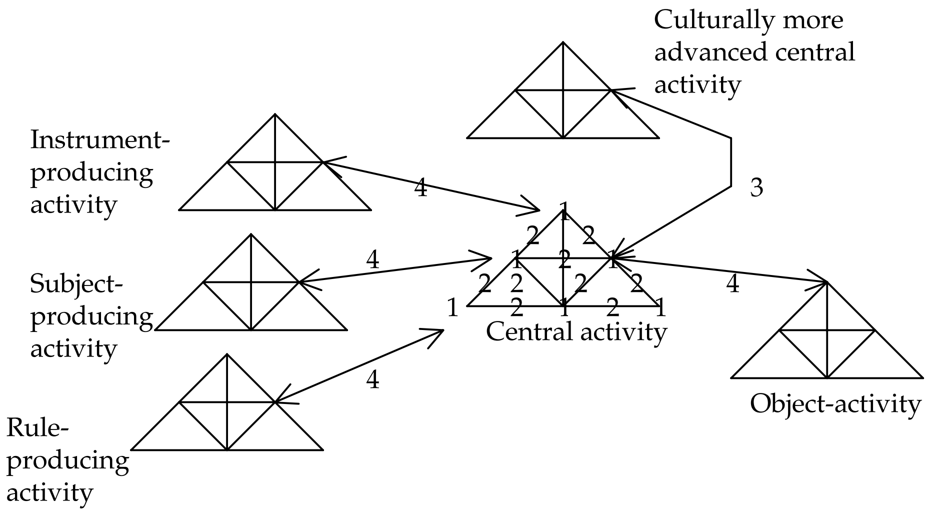
FIGURE 20 The structure of human activity (Engeström 1987a, p. 78)

This systemic model is according to Engeström the simplest possible. Each sub-triangle can be seen as an activity of its own: production, consumption, exchange, and distribution. The most important is still the systemic whole (Engeström 1987a).

This systemic whole is linked with 'neighbour activities' (FIGURE 21). These 'neighbour activities' include first of all the activities where the objects and outcomes are embedded (Engeström calls them 'object-activities'). Secondly, there are activities that produce the key instruments for the central activity ('instrument-producing activities'). Thirdly, there are activities like education and schooling of the subjects ('subject-producing activities'). Fourthly, there are activities like administration and legislation ('rule-producing activities'). Naturally, there are other central activities which are related or connected to the examined activity in some other way, in longer or shorter period. (Engeström, 1987a)

3.2.3 The dynamics of an activity: contradictions as sources of development

According to the activity theory, contradictions are the driving forces leading to changes and development of an activity. Going back to Marxism, most essential property has both an exchange value and a use value. This double nature also penetrates every action and relationship, being an internal source of all contradictions in activities. '*Internal contradictions find their outward expressions in external ones*'. Engeström (1987a, pp.87-89) separates four levels of contradictions (FIGURE 21).



Level 1: Primary inner contradiction (double nature) within each constituent component of the central activity
Level 2: Secondary contradiction between the constituents of the central activity
Level 3: Tertiary contradictions between the object/motive of the dominant form of the central activity and the object/motive of a culturally more advanced form of the central activity
Level 4: Quaternary contradictions between the central activity and its neighbor activities

FIGURE 21 Four levels of contradictions within the human activity system (Engeström 1987a, p. 89)

Korpela (1994) finds Engestöm's emphasis of internal contradictions as the source of development in a work activity restricted. He points out that through facing external threats and opportunities inner shortcomings become reality and an alternative way can be identified. However, it is the inner state of affairs which determines whether an external relation becomes a crucial threat or opportunity or not (Korpela 1994, Engeström 1991).

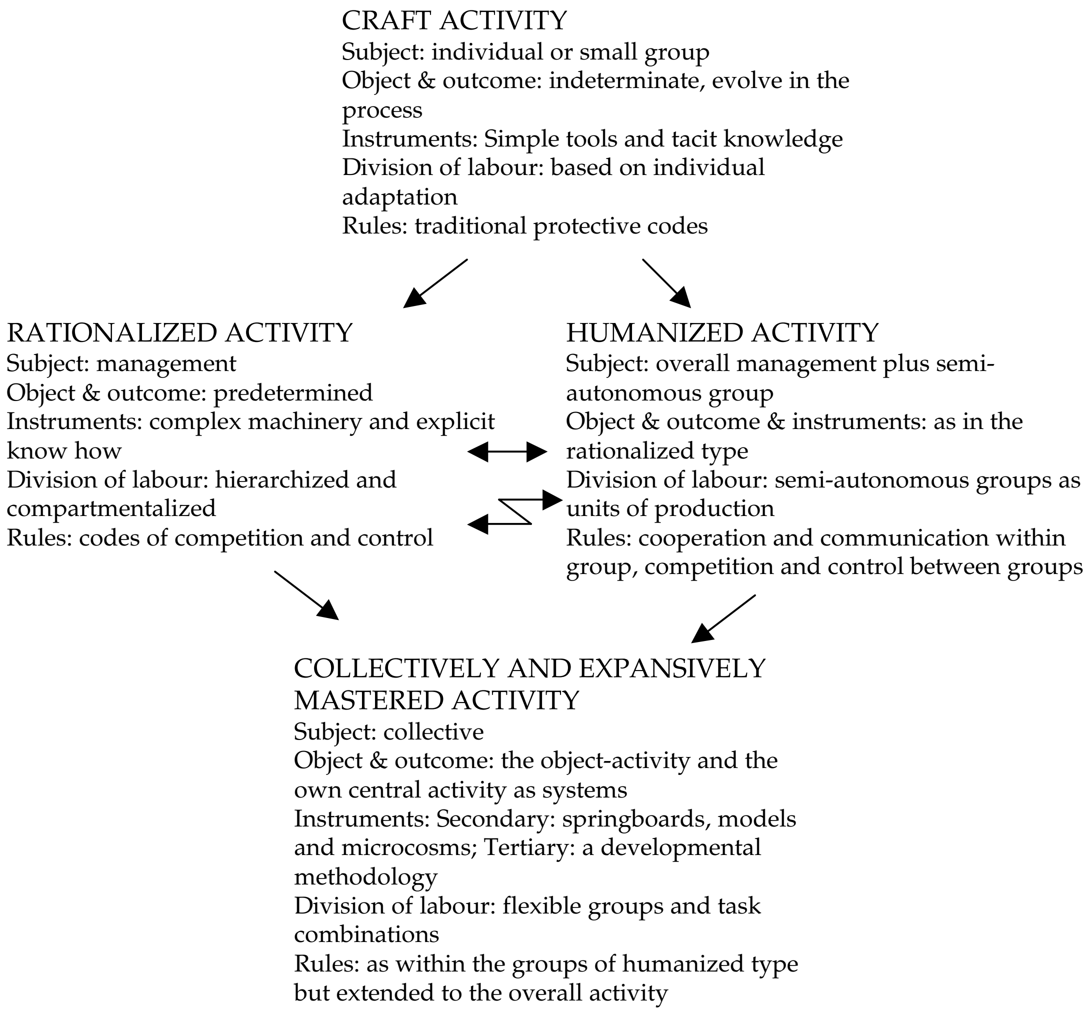


FIGURE 22 Four historical types of activity and expansive transitions (Engeström 1987a, p. 284)

As mentioned before, the historical aspect in work or activity development is emphasised in the activity theory. The development of work in general is characterized with the help of two variables: the degree of complexity and degree of centralization (Engeström 1990b).

I am not going to go into the details of the history of work development here (see e.g. Kuutti 1994, Korpela 1994, Kuutti & Molin-Juustila 1998), but the starting point is a classic *craft activity* with minimal specialization of labour. In craft activity the complexity is low and the centralization is high. The development after that faced with *rationalized activity* with increasing complexity and high centralization (industrial factory), and with *humanized activity* with increasing complexity and decreasing centralization (semi-autonomous industrial work). These two are contemporary and counterpart types of activities. The last type of activity in historical development of work is *collectively and expansively mastered activity*, with high complexity and low centralization. At that level the subject of activity is collective with flexible groups and task combinations, communicative networks, and inherent cognitive of the object and outcome. Kuutti and Molin-Juustila (1998) call them network organizations. The historical types of activity and expansive transition is illustrated in FIGURE 22.

Instead of asking only 'how', work communities start also ask 'why', 'from whom', and 'where to' questions. As a consequence work teams are engaged in modelling and reconstructing their own entire activity systems. This implies that high-level theoretical and conceptual instruments are collectively developed and employed as part of the everyday activity. (Engeström 1987a, 1990b)

One reason why the activity theory has gained popularity within different disciplines is the flexibility of the basic concepts and the flexibility of the level an activity can be observed. The activity theory lets us study the fundamental dialectical relations between the development of the individual and the society in which the person exists (Bødker 1997). On the other hand, it also means that it is in fact impossible to make a general classification of what is an activity, what is an action etc. because the definition is totally dependent on what the subject, object etc. is, in particular situation. The scope of this pattern expands still further when we consider the activity of collective subjects, e.g. a situation in which a group of states is using the United Nations as a tool to preserve peace in the world, which is again an activity. (Kuutti 1991)

3.3 Developmental Work Research (DWR)

In the 1980s a group of Finnish educationists, psychologist, and a sociologist applied the Activity Theory and developed a methodology for promoting work activities (Engeström 1987b). The development of the methodology has been based on practical experiments and theoretical reflections. In his dissertation Engeström (1987a) explained with the help of cases how he has ended up to the methodological cycle of expansive developmental research (FIGURE 23), based on Vygotsky's concepts. I am not going to cover all the steps in this book, just a short overview of the methodology.

The first step, *phenomenology and delineation*, aims to get an insight into the nature of the activity and delineates it. The need state and contradictions of an activity can be accomplished by reading, participating, observing, discussing, and so. (Engeström 1987a)

The next step consists of a rigorous *analyses of the activity*. The object-historical analysis aims to identify and analyse the developmental phases of the activity. It also aims to recover the inner contradictions giving rise to the transition, the development. The theory-historical analysis aims to recognise shared secondary artefacts utilized in developmental phases of an activity. These cultural artefacts are embodied in different modalities, like handbooks, working instructions, fixed procedures etc., but they are all like public knowledge. The actual-empirical analysis aims to describe how the work is practically done, how actors interact and communicate and what are the organizational structures. The output should be evaluated through object-historical and theory-historical analyses. (Engeström 1987a)

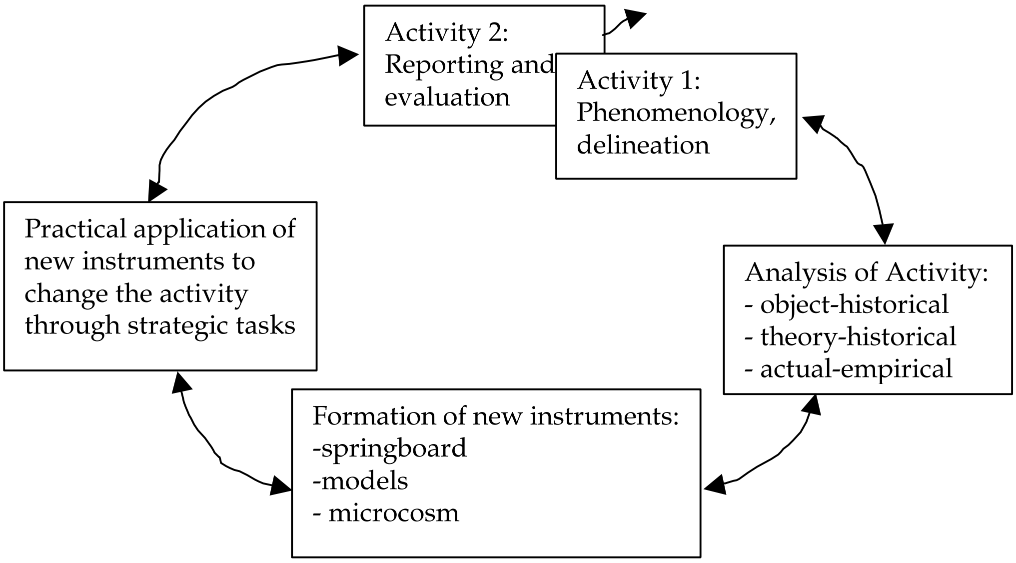


FIGURE 23 The methodological cycle of expansive developmental research (Engeström 1987a, p. 323)

The purpose of the next step, *formation of new instruments*, is to push the participants of the activity to formulate qualitatively new models. The step consists of three elements. Springboard means (briefly, see Engeström 1987a, p. 287) a facilitative technique or so transplanted from some previous context into a new, expansively transitional activity context. Models are needed (Engeström 1987a, p. 295) to envision and project the evolving object and motive of new activity. Microcosms (Engeström 1987a, p. 296) are miniatures of the community upon which the new form of activity will be based. It is a social test bench. (Engeström 1987a)

The next step is to *implement the new instruments into practice*. It usually faces intense conflicts between the old and new system, caused for example by fear or resistance within individuals or groups. Researchers should register and support the change, and also analyse the solutions to the conflicts. The final step is to report and assess the outcomes. (Engeström 1987a)

Engeström's model and developmental work research (DWR) have been used in many practical studies in Finland, for example in flexible manufacturing (Toikka et al. 1986, Vartiainen et al. 1991), in cleaning work (Engeström & Engeström 1986), health care (e.g. Engeström et al. 1988, Engeström 1991), and journalism in a national news paper (e.g. Helle 2000). Developmental work research has also been applied in information systems research in Finland. For example Hyysalo and Lehenkari (2001) studied user-participation in the design process and the use of a diabetes database in two health-centres.

When studying, for example, the work of general practitioners in a health centre, Engeström (1990b) gives three methodological principles, which have been discussed already earlier in this chapter, namely

1. the entire activity system as the unit of analysis,
2. historicity as the basic of classification, and
3. inner contradictions as the source of change and development.

These are the very basic activity-theoretical principles. Typically for these activity-theoretical studies, the 'entire activity system' that researchers are interested in means that the basic structure of human activity by Engeström (1987a, see FIGURE 2 in page 77); object – subject – community, and mediating elements. It incorporates the *object-oriented productive* aspect and the *person-oriented communicative* aspect of human conduct. Thus the approach focuses inside the activity, not outside the activity reaching the organizational boundaries and society around it, at least not theoretically and systematically. The socio-cultural concept, which is emphasised in the activity theory, is restricted to the elements of the structure of the human activity triangle. O*rganizational* and *societal* factors are usually equally important when trying to understand the development of an activity as well as contradictions within it.

There are some very comprehensive methods for work development taking the different perspectives of work activity into consideration quite broadly. For example the Institute of Occupational Health in Finland has developed a method for modelling work processes. It is a very strict method with several matrixes to work through by the practitioners, having also intra-organizational level of analysis in some sense (Leppänen 2000). Historically its development is based on very complex processes in the paper industry, and thus the model is quite complex and structured for such work processes. It is not very open-ended and concentrates mainly on the inside of the activity and its nearby activities.

3.4 Activity Theory, DWR, and Information Systems Research

The general Activity Theory has been applied somewhat in information systems research. Bertelsen and Bødker (2000) summarise that historically speaking, the works of Kuutti (1991) and Bødker (1987/1991a) by using the activity theory in ISD, gave the early impacts primarily in human-computer interaction (HCI), together with Liam Bannon (Bannon & Bødker 1991, Bannon & Kuutti 1993). Since that the use of the activity theory has been expanded to other fields in IS research like computer supported cooperative work (CSCW) (e.g. Kuutti 1994), usability studies (e.g. Vrazalic & Gould 2001, Spinuzzi 1999), participatory design (e.g. Hyysalo & Lehenkari 2001), and user interfaces (Bærentsen 2000). Work in line with human activity has been reported for example in a book by Nardi (1996). The field of information systems is very interdisciplinary. Understanding and design for IT in use demands from the researchers and practitioners a sensitivity towards technical as well as psychological, social, and further matters. Thus the activity theory has become a theoretical tool for ethnographic studies, participatory design, and also psychological approaches. Many studies utilize ethnographic methods like video taping and observation.

An example of usability studies is research that is done in the Activity Theory Usability Laboratory (ATUL) at the University of Wollongong, Australia. The laboratory aims to employ an innovative methodology for usability evaluation particularly suited to highly interactive and complex activity systems. Their methodology makes use of the elements of the activity theory. It was formally established in October 2000 to carry out usability testing of software prototypes, websites, and information systems that support practical human activities, either as individuals or in groups. (Vrazalic & Gould 2001)

In the research of human-computer interaction, Bødker (1987) has applied the activity theory when studying computers and interfaces in human practices. In her thesis she used and refined and enriched the activity theory to understand user interfaces and their design. She explores the possibilities to offer a new conception of user interface with the help of Leontjev's explanations what human beings do when they operate artefacts. The user interface can be seen both as an artefact in user activity, and as an object of a design activity. She has been interested in IT as an artefact and has been studying the computer as a mediator of human activity (see Bødker 1997).

It is not common to use the concept of *work* as a basis unit of analysis in information systems research. There are still some studies of using the activity theoretical approach for studying *work activities* in information systems research. I have already mentioned Helle (2000) in relation to developmental work research studies. She studied the work of journalists in a national news paper by analysing the network of activities focusing on the same object: a daily newspaper. In that case the unit of analysis was a journalists work, and the problems after the installation of a new pagination system. The analysis includes several supporting activities with its' own perspectives for a shared object; owners and shareholders, advertisers, newsroom, middle management, systems department, readers, advertising department, and production department. They used a method called *Change Laboratory* (a laboratory method for work development developed by the Centre of Activity Theory and Developmental Work Research at the University of Helsinki) to analyse disturbances and to transform these disturbances to contradictions into the triangle form. As a result they achieved a broad picture of journalist work and the main problems in this case.

Kuutti (1994) describes the problem situation in the health care sector, where a client had multiple problems – some biomedical diseases, mental problems, and social problems. Thus there were separate, but overlapping and potentially conflicting areas of interest of different helpers (physician in the health care unit, hospital physician, and social worker). The situation was analysed by using a work support typology and the potential areas in which IT could potentially be most useful. The successful result was that a new community and a new activity began to emerge, including all the helpers as subjects, a new division of labour, new rules and new artefacts (like e-mail) as a means of tools.

Helle's (2000) and Kuutti's (1994) cases are examples of using the activity theory in order to study a web of activities. The tension or contradictions between the activities are the source of change. In the same way the activity theory allows you to pay attention to the technical solutions that cross boundaries between activities. One example is Engeström's and Escalante's (1996) study of Postal Buddy. It is a story about the rise and fall of a free-standing electronic kiosk, which was supposed to enable immediate, online services for consumer use. They say that the study of activity networks is only beginning within the activity theory.

Another example of using the activity theory in order to study work activities is by Hyysalo and Lehenkari (2001). They developed an activity-theoretical method for studying the effects of user-participation in ISD through a case study of a design process of a diabetes database. They studied the designing and user-collaboration of the Prowellness Diabetes Management System (PDMS) starting in 1999, after which the system has been introduced into several health-care organizations in Finland. They emphasised historical data consisting of documents, program versions, and interviews related to the PDMS. They also gathered ethnographic data about the user-producer interaction and the use of the system. They conducted research-driven interventions in the on-going user-producer interactions.

The cases presented above are based on Engeström's structure of human activity and the elements within it. What is typical of the activity-theoretical studies in general, is that these cases represent studies where the central activity and the nearby activities are taken into consideration, leaving for example a broader intra-organizational or inter-organizational levels out of the case, not to mention the societal level. There is another feature in the Engeström's model we consider rather restrictive. The analysis of the networks between providing or supporting activities and object activities concentrates on the contradictions between these activities. We like to emphasise that *networking* and *communication* between these activities is worth considering and also mediated by some means.

Bertelsen and Bødker (2000) notice that most studies forget the heterogeneity of different conditions that various groups have for participating in activities of design and the use of information systems. There are some examples of these kinds of studies, for example Korpela et al. (2000a) report their experiences in the early phase of the systems development in Nigeria and Finland concerning health care informatics. Spasser (2000) presents a collaborative work when creating FNA database (Flora of North America) by employing the activity theory as a framework to conceptualise the design, use, and evaluation of a computer-based tool.

Fitzpatric (2000) discusses work practice boundaries in dynamic proximity relationships. She combined the activity theory with Strauss' Theory of Action (1993). Instead of defining boundaries as edge conditions, she defines "a metaphor of boundaries as centre and peripheries, where the effects of boundaries is given by peripheries in proximity relationships to the centre" (Fitzpatric 2000, p. 122). In case of work practice, the centre can be characterised by distinguishing a shared object or purpose. Work practice peripheries can then be defined with respect to this centre. For example, the social world that exists around the care of a patient in a hospital includes firstly, health informatics and equipment and nursing, then management, family, pathology, radiology and consultant, then hospital administration and ancillary services, and finally medicare and health planning as a further periphery. Multiple peripheries in a social world can be characterised by *multiple dimensions*: a degree of commitment, limits of communication and levels of participation. She studied these boundary crossings by means of electronic communication. She concludes that work practice boundaries do not exist in isolation but in the context of multiple boundary forms, which are mediated and interacting.

Concerning central and peripheries I agree with Fitzpatric (2000) that the distinction between what is internal and what is external is often blurred. I like her extensions within the studies of work practices and especially the idea of multiple dimensions, even if the words *centre*, *boundary* and *periphery* still somehow restrict the attention to central activity.

Besides the central activity with its peripheries we believe that extending attention outside of the central activity in general is important in order to understand the contextual and socio-economic factors that have influenced the development and operation of an activity, and the network of activities. For example, economic relations between organizations along a chain of activities as well as the economic and political development in the society in which the activities are embedded, are important contextual factors that affect the inner development of the activities (Korpela et al. 2001b).

The activity theory has also inspired information systems researchers to analyse information technology (IT) as an *artefact* (e.g. Bødker 1987/1991a, Christiansen 1996). Information systems development provides its object activities with new IT tools (artefacts), but it also uses different kinds of IT tools when developing systems, such being an activity itself. In the following I briefly discuss IT as an artefact in user organization and within information systems development. Artefacts are things that mediate between subject and object, or between subject and another subject. As artefacts are not objects of an activity, the subject or user does not normally conduct actions towards the artefact. Like we do not use a hammer without the real object in mind – hammering the nail into a piece of wood. "*To the users, artefacts are what they are meant for*." (Bødker 1987, p. 36)

3.4.1 Information technology in use – IT as an artefact

Thinking of the use of information technology (IT) within an activity, the natural position of it is as a tool in that activity, including both computers and the software running in them. Thus IT as an artefact has a mediating role in an activity in question.

Traditionally artefacts – we can also talk about technology – have been developed slowly during history and have been subject to change. Computer based artefacts have been developed much faster. They are usually quite complex, and extraordinarily, they have usually been designed by people who are not experts of the activity artefacts are meant for, and they are not directly members of that community of computer based artefacts which they are meant (Bødker 1987). This makes information systems quite an extraordinary instrument and thus the understanding of the user community the most important task in systems development. In addition, information systems as artefacts provide more flexibility, compared to more traditional artefacts, in terms of time and place when using the system. Thus shared understanding of an object and the outcome between designers and users would be the prior pre-requirement.

When illustrating the role of the computer application in use, there have been metaphors used like 'system', 'media', and 'tool' (TABLE 7). A tool mediates the relation between the subject and the material object being worked on (*means of work*). The tool perspective emphasizes the production of outcome. The media perspectives of the human engagement with other human beings through the computer application. A media mediates the relationship between the acting subject and the community of praxis surrounding the subject and the activity (*means of communication*). The system perspective is the birds-eye, control perspective, viewing the human user and computer component as rather equally functioning in exchanging data (compare our definition of *information system* in page 11). The subject is lost in the systems perspective. (Bødker 1997)

TABLE 7 Characteristics of the system, tool, and media perspectives (Bødker 1997, p. 152)

	System	Tool	Media
Why	Planning/control	Material production	Communication
What	Data entry + extraction	Shaping materia	Creating and interpreting signs
How	'Low risk' data entry	Transparency – good access material	Transparency – no disturbance in interpretation

Transparency means that computer applications can mediate the normal use activity without 'being in the way'. In normal use situations our handling of artefacts is done through operations. So we do not use artefacts through deliberate actions. (Bødker 1997)

When an information system is designed to improve a certain activity, its impacts concerning that activity are manifold to include all the elements of an activity. For example, there can be new rules for the interaction within the community, the community itself may change, there can be a new division of labour created, the object of activity may change, and so on. Also, the subject – the user – must change in order to learn to use the new system. Thus, the information system penetrates the activity, it becomes part of the activity and it also transforms it (FIGURE 24). (Kuutti 1991)

Area of support / Role of user	External activity	Internal activity	Relations within community
Passive	Automation of well-defined operations	Control: support for pre-determined actions through pre-determined internal models	Coordination: by establishing a division of labour
Active	Tool: support transformation of material objects	Learning: support for the building of a new model for the situation	Communication: enabling the formation of social intercourse

FIGURE 24 Types of work support in information technology (Kuutti 1991, p. 538)

Introducing information systems penetrates the internal aspect of the activity, and every mediating member of the activity structure. FIGURE 24 aims to classify these intertwining relationships. Examining these multiple penetrations clearly clarifies that the research and development for information systems cannot be separated from research and development for whole work activities. (Kuutti 1991)

Even if the technology has been a subject to change, it does not mean that technology would determine society. Neither does society "script the course" of technological change (Castells 1996, p. 5). Instead, there is a dialectical interaction between society and technology. Technology embodies society and society uses technology. Thus we cannot interpret technology or the use of technology without society, and this also concerns the development of information technology.

3.4.2 ISD as an activity

Information systems development (ISD) has a close relationship to work development since ISD affects work activity by facilitating it with new artefacts and even by changing the mode of operation. Work activity, or a work system, is a system in which human participants and/or machines perform a business process using information, technology, and other resources to produce products and/or services for internal or external customers (Alter 1999, 2001a). It also has a surrounding context and relies on infrastructure (Alter 1999).

Also information systems development itself can be seen as an activity that includes all the elements of 'the structure of human activity'. For example, an information systems development project is a collective phenomenon, having a subject (usually collective subject) who transforms an object to an outcome – to some work which needs to be improved (object activity). The collective subject needs tools or means to carry out the work, like physical equipment and development methodology. The work is taking place in some community, which shares the object. There are rules controlling the work and communication, and a division of labour, project organization, roles etc.

In information system design the design object is the artefact, the intended outcome of an activity. The designing subject is the collection of professional designers and members of the practice. Design activity is mediated by design artefacts, like programming languages, CASE-tools, specification standards, systems development methods, and theories of human motor performance. Design artefacts are boundary objects (Star 1989) since they adapt to different situations of application and at the same time maintain identity, thereby mediating divergent needs and viewpoints. For example in cooperative prototyping the prototype plays two roles, it is the continuously moving object of the design activity, but at the same time it is a design artefact mediating the creation of insights and visions into the new information system. This kind of doubleness is a basic feature of information systems design. (Bertelsen 2000)

What is also double-barrelled in information system design or development is that users and designers are driven by different motives and the object of design does not make sense in the same way for them. Thus design artefacts mediate across heterogeneity by tying involved activities together. (Bertelsen 2000)

As information systems development is very much a work development activity at the same time for an object activity (and also very interdisciplinary), it is important that developers and designers work closely with users. Bødker (1997) proposes an active participation of users and systems developers, or design by doing (Greenbaum & Kyng 1991), and a continuous concern for all levels of future use activity. She (1996) also points out that Engeström's change-oriented approach to work development provides a new conceptual framing of the Scandinavian participatory design tradition, the collective resources approach. Overall, we can say that there are two viewpoints for the '*boundary objects*'; users' view of the artefacts as well as designer's or researcher's. Engeström (1990b) argues that it is vitally important that the actors take the system view and the designers or researchers take the personal view, not totally but switching between multiple views (e.g. personal views are not monolithic, since they consist of alternative historically grounded representations).

3.5 Activity Analysis and Development (ActAD)

In this chapter I describe the method developed by the INDEHLA-Methods project, called Activity Analysis and Development, which is based on the activity theory and DWR. The method has been further developed and elaborated during the project, when it was used as a research framework for studying information systems development work as an activity in Nigeria. The chapter is based on the series of publications concerning the method and experiences of using it (Korpela et al. 2000a, 2000b, 2002a, 2002b). First I will introduce the background of the method, and how it all started.

3.5.1 The background of ActAD

In his dissertation Korpela (1994) takes Engeström's Development Work Research on assessment. First he points out that DWR is both a *theory* about the structure and dynamics of human work activities, and a *methodology* to develop collective work activities. He states that the approach is the best approach so far for conceptualising and developing work processes. In the theoretical aspect he finds some fuzzyness, especially between the subject and the community. In Engeström's *structure of human activity* the relationship between the activity as an entity and the social structure around it has not been elaborated theoretically. Thus the work activity is quite a self-contained entity in Engeström's model (Korpela, 1994). Korpela elaborates the model further by separating the community and the social infrastructure (including rules and division of labour), which Engeström added like wholeness (Engeström 1987a, p. 190). In this book I am not going to go into the details with Korpela's argumentation, but he sees the social infrastructure as a social artefacts mediating the community's relationship with individual subjects, instruments, and the object (Korpela 1994, p. 87). He also simplified the model by ignoring non-mediated relations. By this elaboration he ended up with the three-dimensional model, with a mediated relation between community and instruments (FIGURE 25).

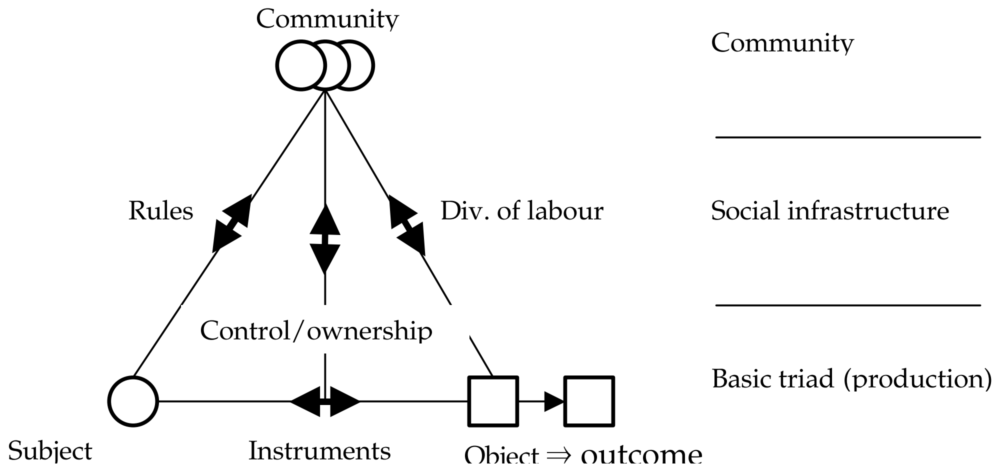


FIGURE 25 A three-dimensional model (Korpela 1994, p. 86)

In the activity theory, the elements of an individual action of work are the subject, the object, the mediating instruments, and the goal. Of course, in most cases, an individual actor does not work alone on the object. The object is shared within a collective work activity. The subjects may use shared or separate instruments on the shared object of work. Thus the subjects are more or less interacting with each other (Korpela 1994, p. 98, Korpela 1999). As Leontjev emphasises, the motive of a collective activity is in its shared object (FIGURE 26).

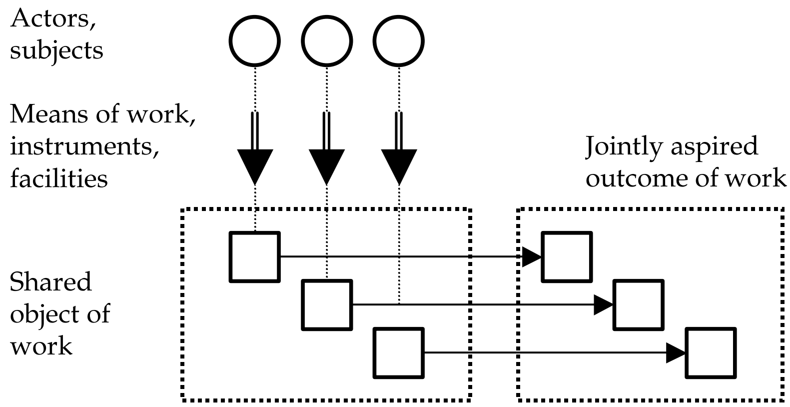


FIGURE 26 Individual actions merging into a joint activity (Korpela et al. 2000a, p. 194)

When several subjects work on a shared object, there needs to be some form of *coordination* between them, and this coordination is mediated by social infrastructure, meaning rules and division of labour (Engeström 1987a). While individual actors act on the object of work through means of work, we can regard that the group taken together as a *collective actor* applies these *collective means* or instruments so that the individual actions are directed to produce a joint outcome. In addition to the rules and division of labour we can think about several other kinds of means of coordination required in order to merge the individual actions into a collective activity. For example, we need a means of communication, the distribution of the means of work among the actors which is governed by the relations of control, and so. (Korpela et al. 2000a)

Further, as Engeström also emphasises, work activity is a *systemic entity*. There must be a relative join between the elements of work activity, a *mode of operation* (FIGURE 27). In general, the misfit or contradictions between various elements, or between the element and the common mode of operation, is the main source of dynamism in the model. (Korpela et al. 2000a)

The egg-shaped model is basically equivalent to the triangle model presented by Engeström (1987a). The contribution here is that the model has been simplified by ignoring non-mediated relations, the dissimilarity of the elements has been highlighted by depicting them by using different symbols, the difference between individual and collective elements has been highlighted by presenting various subjects and their instruments explicitly, the 'social infrastructure' or 'means of coordination' has been elaborated, and the systemic relation between the elements has been underlined by the 'mode of operation'. (Korpela et al. 2000a)

When he continued to modify the theoretical framework of analysing activities, Korpela (1994) refers to Engeström's (1987a) concept of activity networks. Korpela continues by zooming outwards from a single activity to the *network of activities* and *organization*. In general, the outcome of the activity is partly consumed by another activity, and its components parts are produced by some other activity (FIGURE 28).

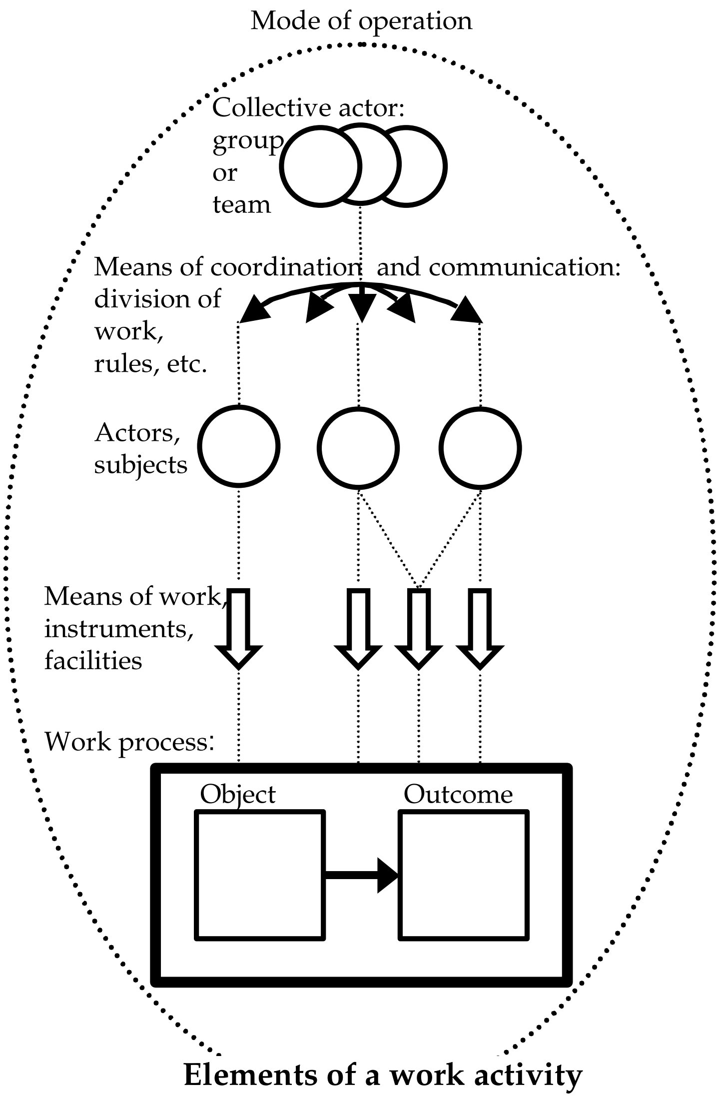


FIGURE 27 Collective work activity as a systemic entity (Korpela et al. 2000a, p. 195)

The so called central activity is just a part of a network of activities (FIGURE 28). The *supported or object activities* are the 'reason for existence' for the central activity, giving requirements for it. The component elements of any activity – objects, means, actors, rules, etc. – are not usually produced in the same activity, but acquired from other activities. The 'raw materials' and tools can be the outcome of some material production activities, while actors can be 'produced' by some educational and reproductive activities, and the rules and other social relations by political or management activities and long-term socio-cultural processes. The *producing or supporting activities* define the central activity's viability or sustainability (Korpela et al. 2000a).

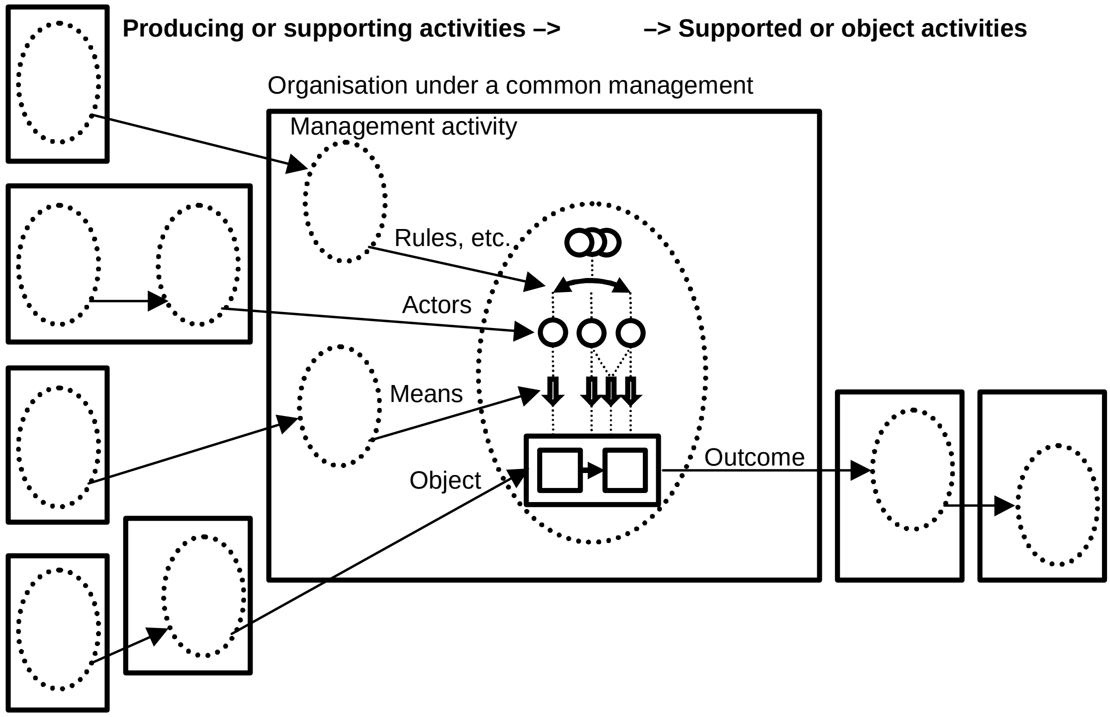


FIGURE 28 A network of activities, split by organisational boundaries (Korpela et al. 2000a, p. 196).

Work activities are embedded in organizations. Sometimes the interaction between activities crosses the organizational boundaries. Thus the management activity not only produces 'social instruments' for other activities within an organization, but also coordination between activities. At that level stakeholder groups are also important to identify.

For the sake of simplicity, FIGURE 28 depicts only one object activity for each activity, and only one producing activity for each component of an activity. In real life, of course, the network of activities is more complex.

The above described framework can be used as a tool or a method for analysing work activities and a network of activities. This framework is useful for a static snapshot, while the real issue is to continue to analyse the *tensions or contradictions* within the current mode, the historical dimension of an activity and further into *cyclic development* of the activity or network, as in Developmental Work Research interventions (Korpela et al. 2000a). Next I introduce how this work development is considered in ActAD.

3.5.2 Structural analysis of central activity

Activity Analysis and Development is a work development methodology. It can be used for several purposes, among practitioners or researchers. Practitioners can be a group of people, a team, who wants to develop its own work practices. Practitioners can also be for example information systems developers, who facilitate other people's work by technology or consultation. ActAD can also be used as a research methodology at the level of work activities. (Korpela 1999)

When starting to use ActAD for the purpose of work development or analysis, the first task is to identify the *central activity*. By looking at the 'shared objects of work' and 'jointly produced outcomes', the central activity can be identified. One must also recognize the people working on the activity. These people involved are the core of the analysis. They are asked to describe the composition and linkages of their own work according to the checklist presented below (TABLES 8 and 9). People can go through the checklist by oneself or together with professional colleagues (as a group). Groups can alternatively discuss their work on their own and then together with other groups. The checklist is in two parts. The first concerns the main constituents of the central activity (discussed already earlier, see FIGURE 27).

TABLE 8 The composition of the central activity (Korpela 1999, Korpela et al. 2000a, p. 203)

1a. *Outcome*: What services or products do we produce?
2a. *Object and process*: What raw materials or prerequisities do we start from? How do we produce the services or products from the inputs we have?
3a. *Actors*: Who are we - what kind of people are needed to produce these services or products?
4a. *Means of work*: What kind of physical tools and knowledge, skills etc. do each of us need for this work? 5a. *Means of coordination and communication:* When we work to produce the services or products, what kind of rules, division of labour, communication etc. apply between us, so that each one's work can contribute to a joint process?
6a. *Group or team*: Putting all the actors together, what kind of a group are we - a closely related team working together all days, a hierarchical organization, a group of people who work occasionally on the same issue but never actually meet, etc.?
7a. *Mode of operation*: Summarizing all that has been said, how can we characterize the way we are conducting our work as a whole - what is the spirit or 'custom of the house'?

TABLE 9 The network of activities (Korpela 1999, Korpela et al. 2000a, p. 203)

1b. *Outcome*: Who needs our services or products ? For what do they need that - to produce some services or products to some others?
2b. *Object*: From whom do we get our 'raw materials' How do they produce what we need?
3b. Actors: Where do we come from - who educates and raises the kind of people needed here? How does that happen?
4b. Means of work: From whom do we get the tools and knowledge we need? How do they produce that?
5b. *Means of coordination and communication*: Who set the rules for us? How are the rules generated? Where do we get the means we need to communicate with each other?

Activities also contribute to other activities, creating a network of activities, as discussed above. It is usually important to look at the wider contexts of the activities, and the level of organizational boundaries is the least that is useful to identify.

The following list of questions in TABLE 8 focuses on the network of activities. The questions aim to identify other activities that benefit from or contribute to the central activity.

After analysing the elements of the central activity, the next step is to continue developing the central activity.

3.5.3 Work development

Activity Analysis and Development follows quite much Engeström's Work Developmental Research cycle (Engeström 1987a, see FIGURE 23) starting with the need state and work analysis and ending with evaluation and a possible new start. Before starting the work on development process there must be a clear need for development. In ActAD, after analysing the central activity's work existing at the present, as discussed previously, the *development must also be analysed*. There are three phases that the people involved and the key stakeholder groups of the relative activities can follow when they want to discuss how the current state of affairs should be improved (Korpela 1999).

1 History. As in activity theory, also ActAD emphasises the history of the central activity: How has this activity and network emerged and developed to what it is now? Can there be identified some phases or stages in the overall development? What is the stage that the activity is at the moment?

2 Problems. Looking at the checklists (TABLES 8 and 9) and figures (FIGURES 27 and 28) presented above, what kind of weaknesses, deficiencies, and imbalances are there concerning the constituents discussed in the questions. It is important to identify the problems both within and between these constituents.

3 Potential. Looking again at the checklists and figures and the constituents in there, what kind of strengths and emerging new possibilities are there within and between these constituents. What would be the new mode of operation of the whole activity? What would be the next step in the historical development? Finally, in order to achieve that stage, what improvements are needed in and between the various parts?

When the potentials and improvements have been identified and reported, the next step is to *implement the work development process*; develop new tools, acquire new knowledge, educate people, improve processes, and relations etc. This is quite a demanding task and needs more detailed methods of engineering, design, and education. Anyway, the important thing is that all the stakeholders are kept involved in the process. The process needs to be *evaluated* and the goals may need to be *redefined*. The outcome of the process can be a starting point for a new round of problem analysis – development – evaluation. (Korpela 1999)

3.5.4 Expanding the framework

So far the framework concerns the activity level of analysis; *intra-activity* level, where the unit of analysis is a work activity, and *inter-activity* level, where focus is on relations between activities. As Walsham (2000) pointed out, most information systems research disregards societal issues and assumes that the results achieved in Western or Northern countries are universal. Thus researchers should cover all levels of analysis in their studies; individuals, groups, organizations, and societies. The idea is to analyse activities *in context*. (Korpela et al. 2001b)

Also Alter (1999) emphasises environment of an activity. He is talking about work systems, and that in order to understand work systems requires an understanding of its environment, including external infrastructure and context. According to Alter (1999), infrastructure means the shared human and technical resources that the work system relies on, outside of the work system. A context means the organizational, comptetitive, technical, and regulatory realm within which the work system operates.

Organisational boundaries are basic elements affecting the seamless operation of the activity networks. From an activity theoretical point of view, we can define an organisation or institution as a group of activities which are coordinated and controlled by a common management activity. In other words, the management activity of an organisation is the primary source for the rules applied within and between the other activities of the organisation. An organisation can be an informal community, a department, a company, a branch of administration, a state or so on. When a producing activity and its object activity are located within the same organisation, the relationship between them is more straightforward than if they belong to different organisations. (Korpela et al. 2000a)

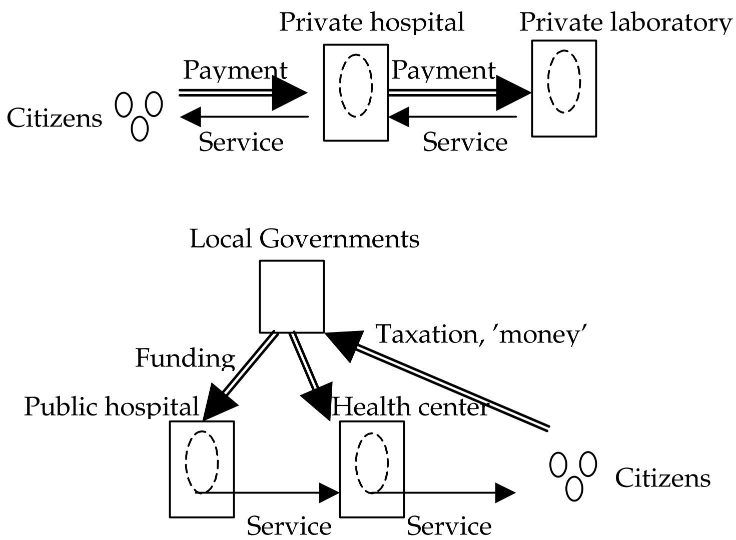


FIGURE 29 Examples of Inter-Organizational Financial and Service Relations (adapted from Korpela et al. 2001b, p. 372)

Boundary-crossing activities are an especially interesting phenomenon. For example, financial relations between organizations do not seem to easily bend into a pure activity framework. FIGURE 29 gives examples of the *inter-organizational* level of analysis. In a market setting, financial relations between organizations follow service relations between activities in the opposite direction, but financial and service chains are not always necessarily identical (Korpela et al. 2001b).

Zooming further from the organizational level, the next step would be the *society* in question. The activity level of analysis must be linked with a societal level in order to adequately understand how such activity networks operate and develop in a given socio-economic context. We need to have an *intra-societal* level of analysis. The unit of analysis is a society, which can be a political entity (state) or a cultural entity (party, ethnic group etc.). The different meanings of the term society should be kept clear: cross-national research is not necessarily cross-cultural, and vice versa. Intra-societal IS research tries to find relations between an IS phenomenon and its societal context (Korpela et al. 2001b). FIGURE 30 illustrates possible elements in an intra-societal level.

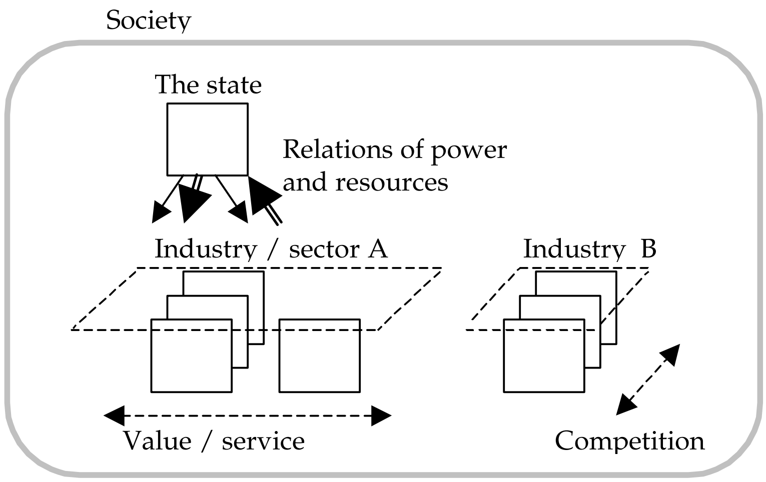


FIGURE 30 Elements of a framework linking organizations with a society (Korpela et al. 2001b, p. 373)

A final zooming outwards will lead to an *inter-societal* level, which includes comparisons between countries or sets of countries. There are two concepts applicable to this level: a service or value chain that spans national or cultural borders, and a grouping of countries into industrialized and developing ones (Korpela et al. 2001b). In his thesis, Korpela (1994) wants to pay attention to the societal level when analysing activities. Here, the history, the environmental context, the politico-economic system etc. comes into the consideration. In FIGURE 31 I have illustrated the broadest and the last level of a modified theoretical framework in Korpela's thesis (1994, p. 214).

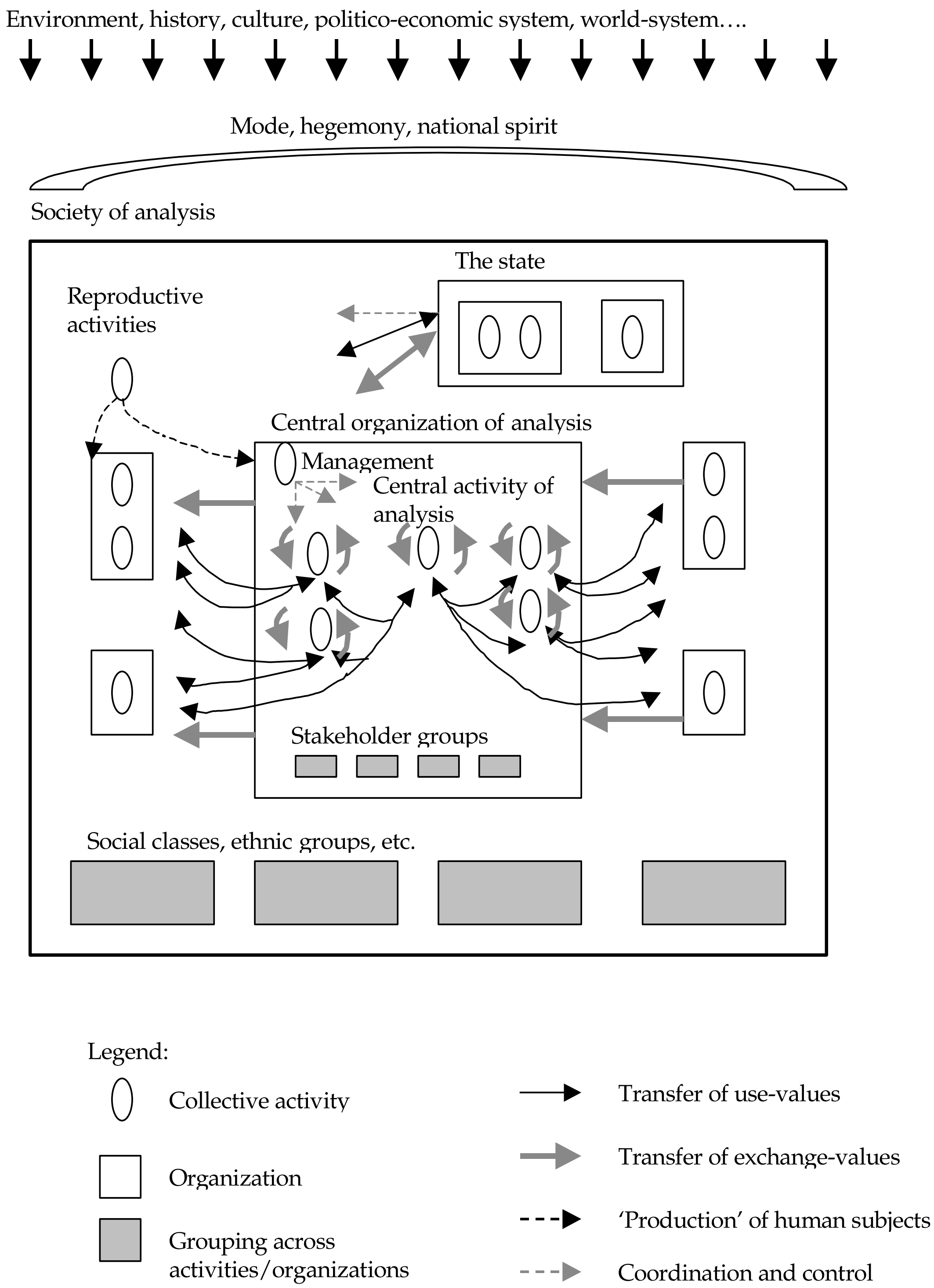


FIGURE 31 The final level of scope: activity, organization, society, supra-society context (adapted from Korpela 1994, p. 214)

3.5.4 Activity Analysis and Development and Information Systems Development

In the following I will describe the information systems development in light of the activities and network of activities, as well as in light of ActAD. The ActAD framework is then adjusted for analysing information systems development as a work activity.

I have earlier emphasised the close relationship between information systems development and work development. ISD process changes and develops work activities by bringing new (kind of) artefacts into the activity and usually changing the mode of operation. An IS developer is an external facilitator who works collaboratively with the client in order to improve the latter's work, just like a work development consultant. When starting to improve work, it is important to take first a holistic view of the activity. Under that are the component elements that need to be improved. IS developers must start from a work development perspective in order to develop the kind of information systems the users really need. Thus in ISD process, IS professionals must also be work development professionals to some extent. This is actually the most important on relatively small-scale development projects which do not allow heavy, long-term, resource-consuming, full-scale work development undertakings. (Korpela et al. 2002b)

ISD within a network of activities. Information systems development is an activity within a network of activities (FIGURE 32). The network of activities in the figure are named according to their 'collective actors', which form the main stakeholder groups in ISD. Activity Analysis and Development can be used in different ways by different stakeholders (Korpela et al. 2000a).

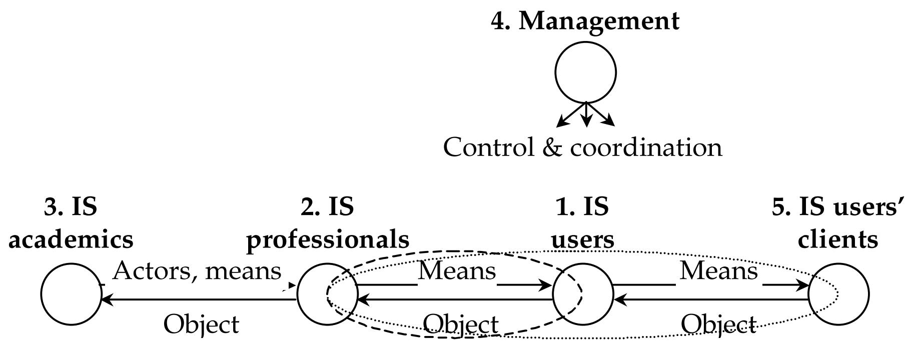


FIGURE 32 The main stakeholders/activities and relations around IS development. (Korpela et al. 2000a p. 199, Korpela et al. 2002b)

The *IS users'* work activity is the 'reason for existence' for ISD. The purpose of ISD is to facilitate this work activity by new information-technological facilities, artefacts (means of work or means of coordination and communication) (Korpela et al. 2002b).

Nurminen et al. (1994) prefer to use the word 'actor' instead of 'user', since the system or the computer cannot be responsible for the software, so the software must be an inherent part of the current user's work. It is the actor who awakes the 'sleeping' technology to be useful (Nurminen et al. 1994). Users – or actors – can use ActAD as a methodology to analyse and develop their own work activity. If this analysis results in a need for improved information-technological facilities, IS professionals are invited to participate in the process (Korpela et al. 2000a).

The *IS professionals'* work is to identify, design, construct, install, and modify the technological facilities for the users. Traditionally ISD was more or less the IS professionals' job – they meet users in the beginning in order to 'capture the requirements', and then delivered the finished 'system' to them after some time. In a more recent, participatory or cooperative way, a *temporary multi-professional activity* should emerge between the IS professionals and users during an ISD project (the dashed-line ellipse in FIGURE 30, cf. Bødker 1996). IS professionals can use ActAD in two ways. Firstly, they can use it for analysing their own work practices or activity (IS development) like any other practitioner. Secondly, they can use it as a methodology to analyse the IS users' work in cooperation with the users. In that case IS professionals work as facilitators in a work development process, specialising in design improved IS facilities. (Korpela et al. 2000a, Korpela et al. 2002b)

IS academics 'produce' professional actors through education and some of the means (ISD skills and methodologies) to the development activity through research activities. IS practice is the 'reason for existence' for the IS academics. IS academics can also use activity analysis and development in two ways. They can use it as a methodology to analyse and develop their own work practices (education and research). They can also use it as a methodology to analyse IS practitioners' work in cooperation with the practitioners. Thus it can also be used as a research methodology. (Korpela et al. 2000a, Korpela et al. 2002b)

Management activity provides the specific rules for activities, like the means of coordination, communication, and control. In addition, there can be more generic rules provided by legislative activities, professional bodies, cultural heritage, etc. (Korpela et al. 2002b)

Finally, the *IS users' clients* provide the 'reason for existence' for the IS users' work activity. The outcome of the IS users facilitates the work or life activities of their clients (customers, patients, etc.). Thus the scope of the participatory or cooperative ISD should be expanded to involve the users' clients as well (another dotted line in FIGURE 32). For instance, in healthcare, we should not just look at how doctors and nurses can work more efficiently, but how the patients can receive better healthcare services through new IS facilities. (Korpela et al. 2002b)

The discussion above refers to participatory or cooperative ways of working between the IS professionals and users during an ISD project. When analysing software developers (software companies, departments or so) and their relations with clients, the core of IS development can be seen as a temporary activity at the border of two organizations. We can see ISD as a boundary activity, as illustrated in FIGURE 33 (Korpela et al. 2002b).

The object of the ISD activity is a problem in the users' work process – a need for better facilities. The clients' activity implies what kind of facilities the users will need. The ISD activity gets its actors from both organizations or activities – the users' and the IS practitioners' ordinary activities. The rules of an ISD activity also come to a large degree from two different management activities, presenting different organizational cultures. The various means of the ISD activity tend to originate from the IS professionals' side. (Korpela et al. 2002b)

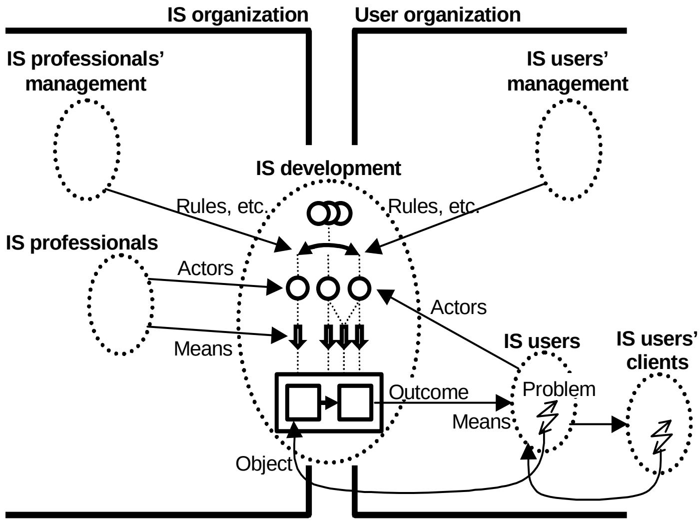


FIGURE 33 The composition of IS development as an activity. (Korpela et al. 2002b)

In Chapter 5 I illustrate how we have applied this framework in our research of ISD in Nigerian software companies.

3.6 Summary

In this chapter I have introduced and discussed the basic ideas of the activity theory and related research done by using the activity-theoretical approaches as a research framework. I have also presented the framework called Activity Analysis and Development, and how it can be used when analysing activities on different levels. Besides the activity level, ActAD emphasises the importance of paying attention to organizational and societal levels when analysing specific activity or a network of activities. Accordingly, the main difference between ActAD and other activity-theoretical approaches is on the levels of analysis, since most other methods are restricted to the activity level of analysis.

Thus we argue that it is important to relate to an activity or network of activities to the society around them. By a socio-cultural context Engeström (1987a) means more or less the elements in his triangle and the relationship between supportive and object activities. Also in the Engeström's model the analysis of networks between providing or supporting activities and object activities concentrates on the contradictions between these activities. We like to emphasise that networking and communication between activities is also mediated by some means. Thus our contribution to the activity-theoretical approaches with the expanded framework is to extend the attention from inside of the activities to the outside of the central activity.

Of course, one can ask whether it is reasonable or even possible to conduct analysis on all the levels – activity, organizational, and societal. It is clear that all the levels cannot be equally covered throughout in one study, the emphasis is usually on one or two levels at the same time. To analyse all the levels needs support from different theories or frameworks. However, we argue that researchers should always identify their specific research scope and context on all four levels to allow other researchers to assess the peculiarity or wider applicability of the study. In addition, if a study applies an intra-viewpoint on any level, the validity of the results is increased if a discussion from the respective inter-viewpoint is included. (Korpela et al. 2001b)

I used the framework of ActAD as a research methodology when I studied information systems development in Nigeria. Based on this approach, I would like to continue with the definition of IS and ISD accordingly (c.f. chapter 2 definitions). An information system is a human activity system. Thus, what makes information technology 'appropriate' is the collective human activity within which it is utilised. The information system is a special case of a work system, or it may be an important part of a work system (Alter 2001b). We define *information system* as (Korpela et al. 2000b, p. 137):

> *the use of information technology (manual or computer-based) in a collective work activity, either as means of work or as means of co-ordination and communication.*

Accordingly, we define *information systems development* as the following (Korpela et al. 2000b, p. 137):

> *Information System Development (ISD) is the process by which some collective work activity is facilitated by new information-technological means through analysis, design, implementation, introduction, and sustained support.*

I am also going to use the ActAD framework as a model when analysing my cases in this thesis. It must be noted that in many levels the framework is still quite tentative. The activity level is tested and elaborated actively during the research process. The organizational level is a very traditional level of information systems and information systems management research. The most challenging level is the societal level. Further development and research is needed for this activity analysis framework especially to relate the activity level to the societal and inter-societal levels of analysis (c.f. Korpela et al. 2001b). So far the methods on organizational and societal levels are based on intuitive understanding of researchers, but there is a need to connect more rigorous methods to the analysis.

4 DEVELOPING COUNTRIES, AFRICA, AND NIGERIA – IN TERMS OF INFORMATION SYSTEMS

This chapter introduces the research context in more detail. The focus is mainly on Western Africa, since the study is taking place in that part. The history of the continent is presented briefly, since it is important to know about the past in order to understand the present and the future. The concept of development is clarified. The literature review consists of IT in Africa in terms of goals and challenges, and ISD in Africa. The appropriateness of IT for development goals is discussed. Nigeria is presented in the light of the research questions.

4.1 Africa – and the concept of development

Historical view in brief. The history of Africa is long and rich in every respect. In many parts we do not know much about it, since there are few written texts, especially from eastern and southern Africa. However, the development and history has been very much related to nature. As far as we know, the environment in Africa has been hostile because of ancient rocks, poor soils, fickle rainfall, abundant insects, and a unique prevalence of diseases (e.g. malaria, hookworm anaemia, syphilis, leprosy, trypanosomiasis). Economically, the history is also a continuous dialogue between geography and history – form the very beginning of agriculture to the introduction of modern industry (Hopkins 1973).

The story begins with the evolution of the human species in eastern and southern Africa, whence it spread to colonise the continent and the world, adapting and specialising to new environments until distinct racial and linguistic groups emerged. The human evolution started between 6 and 4 million years ago, when the hominids (*Australopithecines*) separated their closest animal relatives. Early *homo sapiens* emerged some 400,000 years ago and tool making 'wise man' about 40,000 years ago. Based on the stone stools and the mitochondrial DNA tests, research encourages the theory that Africa's first anatomically modern men were the ancestors of all modern human beings. (Iliffe 1995)

Until the later twentieth century Africa was an underpopulated continent. Its societies were specialized to maximize numbers and colonise land. Ideologies focused on fertility and the defence of civilization against nature. Polygyny was supported by social organization. Scattered settlement and huge distances hindered transport and trade, even if there were highly organized market systems for example in Western Africa. Many of the largest population concentrations remained entirely stateless. (Iliffe 1995)

Among the most momentous times in North Africa were the third and fourth centuries when there was a spearhead of *Christianity* and 800 years later, when it was the pivot of *Islam* and a commercial network encompassing most of the Old World. There was demographic growth during that time, but it was destroyed by the Black Death during the fourteenth century for 500 years. Trans-Saharan trade grew very swiftly in the early Islamic period, although there had been some trade before that. The opening of the Atlantic coastline of Africa to European shipping was in the fifteenth century. The trade included for example slaves, cloth, pepper, and ivory for copper and brass, coral, and European textiles. Around 1400 there was an emergence and steady growth of state systems in Africa, as well as in Europe. (Oliver & Atmore 1981, Iliffe 1995)

The Atlantic slave trade. T*he Atlantic slave trade* started in 1441, and it has been estimated that the number of slaves during the years 1450-1900 has been more than 10,000,000. The slave trade peaked during the eighteenth century and then declined slowly during the nineteenth century. The most important change took place during the mid-seventeenth century, when the Dutch destroyed Portugal's position along the West African coast. British and French traders gradually supplanted the Dutch. Finally, England was the lead supplier. 42 per cent of exported slaves went to the Caribbean, 38 per cent to Brazil, and fewer than 5 per cent to North America. (Hopkins 1973, Iliffe 1995)

Due to the struggle to build up the population had hitherto been African history's chief theme, the slave trade's demographic impact was potentially high. We don't know how severely the slave trade affected western Africa's demographic history. At that time the arrival of the American corps, maize and cassava, made more food available in a region of poor nutrition. On the other hand, the Atlantic trade also exposed new diseases, for which West Africans had little resistance (e.g. tuberculosis, bacillary pneumonia, plague, venereal syphilis, and smallpox. (Iliffe 1995)

West Africa's foreign trade as a whole expanded dramatically during the first half of the nineteenth century, for example with Britain and France. Pre-colonial mining (e.g. gold, salt) and manufacturing (e.g. clothing) were significant. Between the fifteenth and nineteenth centuries almost every part of the continent was drawn into a world economy dominated by Europe. Import of European cotton cloth multiplied about fifty times. The most valued import was guns. The effects must not be exaggerated, since domestic production and consumption remained its economic core. Trans-Saharan trade saw its golden age before the end of the sixteenth century, but it survived and its value actually increased during the nineteenth century. (Hopkins 1973, Iliffe 1995)

By 1880, three independent systems of trade in Africa had emerged – the Atlantic, the Mediterranean, and the Indian Ocean cost systems. However, there were hardly any routes linking these systems. It was not until the nineteenth century that all these trading systems were linked together by both land and river. (Boahen 1989)

The effects of the West African external trade, particularly the slave trade, are believed to have retarded the economic development of Africa, and may even have prevented the continent from achieving an indigenous industrial revolution. On the other hand, international trade undoubtedly brought benefits, at least to some of the parties concerned. However, the external trade failed to act as a leading sector, but so did the international African trade. It failed to act as an engine of growth in Africa. The export sector, besides being comparatively small, established few beneficial links with the rest of economy. Towns and large villages were a common feature of the West African landscape in the pre-colonial era, even if the habitants were mainly agriculturalists. The most important economic unit in virtually all West African societies was, and still is, the household. Slave labour was present in West Africa long before the rise of the Atlantic trade. It was a long and varied institution, a political function. Slaves were usually fairly specialised workers, needed when domestic exchange activities created employment, which could not be met by local labour. (Hopkins 1973)

All in all, the Atlantic slave trade, and the slave gathering, was dramatic and a horrible event in African history. The most important effect is perhaps psychological and is focusing on humanity and human development, which means for example the enlargement of relevant human choices (Esteva 1992). The institution of slavery was abolished in all of Britain's imperial possessions in 1833, and in 1848 slavery was abolished in all French colonial possessions. (Hopkins 1973)

Colonialism. The next momentous times in Africa were faced during *the colonial invasion*. During the last twenty years of the nineteenth century the European Powers swiftly and painlessly partitioned the map. To implement the partition on the ground was anything but swift or painless. Widespread possession of arms, codes of military honour, and a long hostility to governmental control made popular resistance to conquest formidable. The colonial impact varied dramatically from place to place. They differ because colonial change was contradictory and subtle. (Iliffe 1995)

The most surprising aspect of the imposition of colonialism was its suddenness and its unpredictability. The majority of African states and policies were enjoying their sovereign existence, and their rules were in full control of their own affairs and destinies. By 1880, Africa was in a mood of optimism and seemed poised for a major breakthrough on all fronts; a new and modern Africa was emerging. The most inhuman and abominable slave trade had been replaced by trade in natural products, which has become known in typical Euro-centric terms as legitimate trade. There were attempts to modernize societies, since many African nations were aware of the development and industrial changes in Europe at that time. The modernization included industry, monetary systems, military systems, and also constitutional system and development of schooling, postal services, sanitary reforms and custom duties on export. As important is the intellectual activities that occurred in West Africa in particular and in Africa in general, which the main consequences were Ethiopianism, Pan-Africanism, and the ideology of African personality, and a generation of pride and confidence in the Negro race, Europe had witnessed the industrial revolution and was desperately in need of markets as well as raw materials. (Boahen 1989)

All of Africa, except Liberia and Ethiopia, was seized and occupied between 1880 and 1900 by Britain, France, Germany, Belgium, Portugal, Spain, and Italy. The early colonial period was probably the most destructive in equatorial Africa, where violence, famine, smallpox, sleeping sickness, venereal disease, and influenza coincided. By contrast, the early colonial crisis was less acute in the West African forest. Earlier exposure to European diseases may have given the forest people greater immunity. They also escaped famine or epidemic sleeping sickness and their prosperity from cash crops may have improved nutrition and medical care. Colonial Africa was restructured around new growth points, which were towns, mines, European estates, and African cash-crop farms. (Boahen 1989, Iliffe 1995)

To safeguard the interests of the expatriate communities and in conformity with the racist ideas of the day, discrimination was practiced in one form or another by all the colonial powers – more crude and overt in the French colonies and more subtle and covert in the British and German colonies. To varying degrees all the colonial rules also condemned everything African in the cultural field and tried to produce Africans in their own image. (Boahen 1989)

West Africa was the region in which there were the fewest changes in basic economic arrangements. This was perhaps due to the environmental considerations. West Africa is not particularly well endowed with mineral resources, and the environment also tended to discourage the European immigrant settler-farmer or plantation-estate proprietor. It has a reputation of 'White Man's Grave', especially because of malaria (Munro 1976). During colonialism, Africans were forced to grow certain cash crops, such as cotton, instead of diversity of agriculture. The political economy of colonialism was characterized by the virtually total neglect of industrialization and manufacturing and the refusal to process locally produced raw materials (Boahen 1989). For example, until the 1920s Africa – principally West Africa – was furnishing virtually the whole of the world's supply of oil palm products (Wickins 1986).

Capitalism, urbanization, Christianity, Islam, political organization, ethnicity, and family relationships all took particular forms when Africans reshaped them to meet their needs and traditions. To see colonialism as destroying tradition is to underestimate African resilience. Among the most influential features during the colonial period also include the development of railways and motor transport, education, and health care. All these were very unevenly distributed, and there were no universities until the 1940s. Concerning health care and education, they were not actually mentioned to the African people. Also the metropolitan languages were forced to speak, which are today the official and business languages. In addition, the new social order emerged, meaning social mobility in the traditional African social order. On the other hand, there was the downgrading of the status of women. So, Africa's colonial period was as traumatic as it was brief. The most important consequence of colonial occupation can be seen in demographic growth. Between 1920 and the late 1940s Africa's population may have increased from some 142 million to 200 million. Another impact was the appearance of the independent African states of today. Most of these states are artificial creations, which has caused serious problems, many of which have still not been solved. In addition, inter-African trade ended quite totally. (Boahen 1989, Iliffe 1995)

Liberation. Within a few years Africans became conscious of the iniquities, inhumanity, oppression, and exploitation of the colonial system, and began to react to it. The initial momentum for *liberation* was strongest in the north, starting in 1950s. West African nationalism took a predominantly constitutional form, while in East Africa and Central Africa, by contrast, violence was crucial. When most African countries became independent around 1960, everything conspired to raise expectations. Nationalism aimed to imitate the most modern nation states. Nationalists believed that colonialism had retarded their countries. They knew that their frail regimes depended on rapid economic progress. All had experienced rapid economic growth in the 1950s, when high commodity prices had enabled colonial governments to implement development plans emphasizing infrastructure. Most new states had relatively small public debts, ample land, and free peasants. They were poor states, but not the world's poorest. To expect rapid economic transformation was naïve, but to hope for significant growth was reasonable. And it happened, at first and in most countries. Between 1965 and 1980 sub-Saharan Africa's Gross Domestic Product per head grew at an average of 1.5 per cent per year. During 1980s GDP declined by 1 per cent per head per year. The turning point for Africa came during the later 1970s. (Iliffe 1995)

Until that point economic growth had taken three main directions. One was a continuation of the post-war cash-crop boom. The second growth area was mining, where Africa's chief potential lies. Mining also helped to make sub-Saharan Africa's industrial sector a third growth area. Nigeria's manufacturing sector grew from the years 1965 to 1980 at 14.6 per cent per year. This modest economic success turned into crisis during the later 1970s. Among many reasons, one was uniquely sudden and rapid population growth. To expand services like food, schooling, housing, dispensaries and so absorbed the surplus available for investment before there could be any thought of development. Changes in the global environment were a second reason for crisis. When world oil prices multiplied sixfold during the 1970s, Africa's dependence on motor transport left it especially vulnerable. Africa's terms of trade deteriorated sharply from the mid-1970s. Also the prices of export commodities, like copper and agricultural products, fell. As other continents produced competing commodities and the growth-point of international trade shifted to the exchange of manufactured goods between industrial countries, tropical Africa's share of world trade probably fell to its lowest point for a thousand years. One result was dept. A few countries borrowed recklessly during the 1960s, but the general crisis began with the oil-price increase: between 1970 and 1976 Africa's public debt quadrupled. By 1991 Black Africa's external debt exceeded its annual GNP. (Iliffe 1995)

Debt was the point at which the global economic environment gave way to African policy decisions as the chief reason for crisis. Independent African states had vastly different economic experiences. This was partly because they had different opportunities, like Botswana who have diamonds and the highest growth rate in the world, but the differences were also because their leaders made different economic choices. At independence around 1960 most economists believed that poor countries could best achieve development if their governments extracted resources from peasant agriculture and invested them in more modern sectors. For example in Ghana this strategy proved disastrous. The factories were generally too large and inefficient and the public-sector employment increased excessively, taxation and corruption increased, among other things, thus the result was that the government borrowed from abroad and multiplied the money supply one hundred times between 1965 and 1984. (Iliffe 1995)

Despite an almost complete success at political decolonisation, Africa failed at economic development and economic decolonisation. Politically, many problems were compounded by the haste, sometimes the violence, and, paradoxically, the idealism of decolonisation: opportunistic coalitions, regional rivalries mobilized for political competition, constitutions tailored to short-term ends, expectations inflated by easy victories, and locally minded people exercising universal suffrage. Africans experienced several civil wars, e.g. the Republic of Congo (later Zaire), Angola, Mozambique, Liberia, Somalia, and Nigeria. Such disasters, together with the great responsibility resting upon the leaders of new states, made it easier to understand the jealousy and ruthlessness with which Africa's rulers exercised power. So did the sheer difficulty of political democracy in African circumstances. (Iliffe 1995, Mazrui 1999)

Faced with these pressures, most leaders of newly independent states relied first on bureaucracies inherited from colonial rule. Inflated in size, hugely expensive, and authoritarian, these bureaucracies nevertheless provided frameworks without which many new states would have disintegrated. These forms were varying in degrees of patrimonial, thus office was conferred in return for personal loyalty and service to ruler. Such regimes were held together by personal relationships among a small elite. Each elite member headed a personal clientage, usually on tribal or regional lines, which imposed burdensome obligations but linked him to a locality and supported his claim to be its spokesman and protector. Such patrons might be hereditary aristocrats, educated technocrats, or upstart party bosses. In their 'hegemonic project' to dominate society, ruling elites generally drew on three additional institutions.

One was a single political party, the second was the army and the third, a more reliable buttress, was the international order. Until the Cold War ended in the late 1980s, foreign aid gave African rulers extensive patronage at very little cost in dependence. The price, possibly worth paying, was unresponsive regimes, xenophobia towards other African nationals, and the collapse of pan-African dreams. (Iliffe 1995)

As the state declined, society, always the true strength of African civilisation, adapted to new conditions as it had in the past adapted to the slave trade or colonial rule. One example is education, which multiplied on all levels. Health care was another example, but migration to towns increased. During the 1980s sub-Saharan Africa's townsmen increased twice as fast as its population, forming 29 per cent of the total in 1991. Survival in decaying cities depended heavily on informal occupations, which employed for example some 72 per cent of Nigeria's urban labour force in 1978, including its innumerable women traders and youthful apprentices. Informal occupations merged into the 'second economy' of black-marketeering, smuggling, corruption, and crime. (Iliffe 1995)

The concept of a developing country. The concepts of 'developing countries' and 'industrialised countries' are used in this study to indicate the division of the world roughly into two blocks, rich and poor. The different regions in the world have developed unequally during history. According to Amin (1997), the polarization has become the immanent by-product of the capitalist system in the modern era.

The evolution of capitalism started before the industrial revolution with mercantilist era (1500 – 1800). After the industrial revolution up to the end of the Second World War, the world system was characterized by the classical form of polarization to industrialized and non-industrialized world. After the war, the polarization of the world was divided into three. There was the 'first world', also called as 'north', including western countries. Then there was the 'second world', including eastern countries with socialism. And finally, there was the 'third world', also 'south', including Africa, Latin America, and Asia (e.g. Hettne 1990).

During the post-war period industrialisation also took place in the peripheries, mainly in Asia and Latin America. The post-war cycle has transformed the old system to differentiate a semi-industrialized 'third world' and an un-industrialized 'fourth world' (Amin 1997). However, there is a world system, a world economy, which has expanded steadily and covers most of the world (Hettne 1990). Unfortunately, this world economic system generates inequality and runs on inequality (Lummis 1992). The most recent period, since 1990, has faced new kind of development in the world polarization in terms of globalization (Amin 1997), and also in terms of recent terrorism. TABLE 10 presents some figures indicating this inequality. For example, between 1988-93, around 25% of the world's people received 75% of the world's income, and in 1993, the poorest 10% of the world's people had only 1.6% of the income of the richest 10% (UNDP 2001).

The term 'developing country' has originated from the decolonisation period, replacing the terms like 'backward societies' or 'emergent nations'

(Korpela 1994; Alavi & Shanin 1982). There is another division which derived from Max Weber, namely 'traditional' and 'modern' societies. According to Leys (1982, Korpela 1994), traditional society means that social relationships are determined by birth (like kinship, nobility, serfdom), and in modern society by achievement.

TABLE 10 Basic World Development Report statistics for 2000 (World Bank 2002)

	Nigeria	Sub-Saharan Africa	Low-income countries	USA
Population in millions	127	658.9	2,459.8	281.6
GN income in US$ billions	32.7	310	997	9,601.5
GNI/capita in US$	260	470	410	34,100

Along with world development, an interest in the development of the 'backward areas' rose after the Second World War, also by social scientists, besides anthropologists (compare to *evolutionist* theories of 'barbarian' and 'civilized' comparative method, Hettne 1990). According to Esteva (1992), after President Truman's speech in 1949 of 'underdevelopment', the majority of the world's population were no longer seen as diverse peoples, but as a homogenous mass characterised by their condition as underdevelopment (also Nustad 2001). After the Second World War, according to Järvelä and Kuvaja (2001), development had traditionally been seen closely related to modernisation of the whole society, including economic, social, and political structures (Preston 1986), particularly in industry. Thus at that time development was seen as equivalent to economic growth and communities were seen as passive objects of post-war restructuring (interventionism). This restructuring in Europe was based on the Marshall Aid from US (Hettne 1990). There was a belief in the possibility of linear western modernisation in the Third World (Waylen 1996).

Nowadays, development and modernisation have to be separated analytically, even if they are both societal change processes (Järvelä & Kuvaja 2001). The concept of *modernisation* does not evaluate the nature of societal change, being it negative or positive. In other words, modernisation exists in every society, but its nature varies. The concept of *development* is considered to present a positive social and economic change, as well as policies to achieve it. On the other hand, there can be no fixed and final definition of development, only suggestions of what development should imply in particular contexts (Hettne 1990). Still, development cannot delink itself from the words which it was formed, like growth, evolution, maturation (Esteva 1992). According to Esteva, development is a change toward a desirable goal.

I am using the terms of developing countries and industrialised countries, mainly for conventional reasons. These terms are used in the literature of development informatics in general. I am not referring to any cultural or other internal reasons when speaking about developing countries. Rather, I refer to the global political and economical system of the world, how it has developed unequally and what have been the consequences. These consequences include for example a low income level, low productivity and low domestic demand (e.g. Korpela 1994).

Development theories. The development of development theories has not been smooth. It has been characterized by theoretical contradictions and ideological polarizations (Hettne 1990). Within the development theory, there are three general theories or paradigms: modernisation, dependency and multiplicity paradigm (e.g. Hettne 1990, Nulens 1998).

In the *modernisation paradigm*, underdeveloped regions are expected to close the gap between them and the advanced countries by imitating the latter (e.g. Hettne 1990). The paradigm believed endogenous reasons of under-development, being quite racist (c.f. Korpela 1994). The degree of development is measured by quantitative variables like income and investment. Rostow (1960) describes five stages that all developing societies had to pass: 1. the traditional society, 2. the pre-takeoff society, 3. takeoff, 4. the road to maturity, and 5. the mass consumption society. Western technology is believed to be neutral and an accelerating instrument for development. Culture is considered as irrelevant in that development, rather an obstacle. Culture is supposed to adapt itself to the economic process as well as to technology. In its simplistic form, the modernization paradigm served as development ideology, simply rationalizing cultural colonialism. However, transferring the western model to a third world setting has not been successful. The modernisation theory seems to be too simplistic to solve world problems. This paradigm was followed mostly in the 1950s and 1960s. (Hettne 1990, van Ryckeghem 1992, Nulens 1998)

In the 1970s there was a sharp criticism against the modernisation ideas. The *dependency paradigm* was elaborated in the third world, particularly in Latin America (Hettne 1990). The paradigm believes that capitalism produces a division into core and periphery, the latter being dependent on the former (Hettne 1990, Korpela 1994). In that paradigm the purpose was to overcome dependency, and that was possible by disassociating itself from the worlds economy and by striving for *self-reliance*. That was possible by large import-substituting technology programmes. The first vision on *appropriate technology* was defined: a strategy for breaking dependence and reaching economic growth from within. In that paradigm, culture did not have a clear role. Cultural dependency existed only as a reflection of economic dependency. In general, the focus in the dependency paradigm is on external causes of underdevelopment. National problems were not taken into consideration, even if they were causing development difficulties. (Hettne 1990, van Ryckeghem 1992, Nulens 1998)

Hettne (1990) notices that to some extent the newly industrialized countries (e.g. Brazil, Mexico, Argentina, Chile, Taiwan, South Korea) were encouraged by the capitalist world order, while countries pursuing self-reliance strategies were discouraged or even destablized. However, there is no key factor for success, only a set of variables which are different and dependent on the correct timing. He refers to world system theorists who argue that development is basically a matter of changing the structural position from a peripheral to a semi-peripheral one. This is not accepted by the contemporary Marxist concepts of the mode of production and merchant capital. In addition, neostructuralists consider that the world economy constitutes a structured whole, and its constituent parts display various forms and degrees of dependency. Hettne concludes that the 'failure' of self-reliance must be understood in relation to structural and political changes in the world. The relevance of self-reliance as a strategy, implied in the dependency approach, should be seen as a learning experience of delinking. (Hettne 1990)

It is worth mentioning here one of the most comprehensive and totalitarian policies to modernise the structure of economies in developing countries, namely the Structural Adjustment Programme (SAP) by the World Bank and the International Monetary Fund (IMF). It was implemented in the Third World, particularly in Sub-Saharan Africa, during the 1980s. In short, the aim was to solve the problems in Africa by liberating the market forces and adjusting to the global division of labour (Korpela 1994). These programmes did increase the portion of industrial production in the national economy in many developing countries, but the impact to the well-being of the citizens has been widely questioned (Järvelä & Kuvaja 2001). The SAP has been even claimed to harm seriously the political economy of Sub-Saharan Africa (Järvelä & Kuvaja: Castells 1998). According to Korpela's (1994) review, the SAP not only brought severe hardship for the majority of Africans, but also failed to achieve its own objectives of increased agricultural output, export earnings, or macroeconomic growth. In the SAP ideology, IT is considered to provide the new potential to leapfrog into the information society, without taking the cultural aspects into the consideration (van Ryckeghem 1992). Korpela summarises his examination by concluding that African governments are forced to accept these programmes since they are a condition for international financial arrangements; SAPs are part of the debt crisis of the developing countries.

Järvelä and Kuvaja (2001) summarise that these programmes bound development to the macro-economic measures, linear and equal to all regions, thus disregarding social and cultural differences. Hettne (1990) concludes that development is more than changing positions within an international division of labour, and the production and distribution of material goods. He says that development concerns people, their way of life and their conceptions of the good life, as determined by their cultures.

> *In the 90s, partly due to the incapability of macro-level programmes to produce sustainable well-being to the majority of people in developing countries, communities have been the major focus of development activities and they have been seen as central actors to define contents of development (Järvelä & Kuvaja 2001, p. 20).*

Both the modernisation and dependency paradigms focus on the concept of growth and profit. The third paradigm, what Nulens (1998) calls *multiplicity paradigm*, questions the unlimited growth and the existence of a universal development model. The paradigm can also be called as 'another' or 'alternative' development (Hettne 1990). There are different kinds of development processes in different regions, and they must be evolved bottom-up. The development is based on qualitative principles (by Dag Hammarskjöld Foundation) like basic needs, self-reliance of local communities and participatory democracy. The development is also endogenous and sustainable. Self-reliance includes some questions like the size of a proper economy, the level of self-reliance (local, national or regional), and the degree of self-reliance (e.g. meaningful cooperation) (Hettne 1990). One problematic area also is the rise of ethno-politics, which must be related to economic development. The question is about *ethics*, for example with scarce resources. However, the culture is considered in a broad sense (ethno-development) and technology could be of great help, if appropriate. Van Ryckeghem (1992) defines appropriate technology in this context as small, simple, capital-saving, and labour-intensive. According to her, building proper technologies became the newest strategy for breaking the dependency relation with northern countries. Nulens (1998) considers these objectives rather utopian in the current international power structure. Kitching (1982, p. 180) points out that "*an attractive utopian vision is not an adequate basis for a theory of development, nor does the desirability of a state of affairs guarantee its possibility*". Even if this attractive utopian vision is not an adequate basis for a theory, it should be allowed to be an adequate basis for desired goals. The development needs not always to be based on Western or modern values.

In spite of all these efforts, we have not succeeded in narrowing the gap between rich and poor countries. The 1990s gave birth to a new term of development – *human development*. In 1990 The UNDP published the first Human Development Report, which pays attention to socio-economic development. It combines life expectancy deprivation, adult literacy deprivation and real GNP per capita deprivation. The process of human development means the enlargement of relevant human choices. (Esteva 1992)

A radical reaction to the dilemmas of development can be found from a '*Post-development*' approach. It offers an explanation of why so many development projects seem to fail (Nustad 2001). The approach criticizes development theories as development has been practised since the Second World War (WWII). Nustad summarises some arguments of post-development based on the literature. For example, development was launched by the Americans to forestall the spread of communism. Secondly, development is a unitised homogenous power play with the 'poor' as the victim, especially right after WWII. In addition, the idea of intentional (interventionist) development is inherently based on a notion of trusteeship. Even if the new versions of development argue against a top-down approach to development, the question is still about guiding those who are not developed. The reason is that a development process is initiated with a specific goal, although developers portray themselves as facilitators, they still know where the process ought to be heading. Also, they have to construct the field in such a way that intervention is possible. Thus a local perspective is substituted for a more global perspective (Nustad 2001). So development is rejected not on account of its results but because of its intentions, its world-view and economic mindset (Nederveen Pieterse 2000). Development is considered the same as the westernisation of the world. Escobar (1992) concludes that development is a 'Frankenstein-type dream', or an 'alien model of exploitation'. According to him, the dream of development is over. Nederveen Pieterse sees that a little bit confused idea.

In post-development, concerning the trusteeship, development should grow out of 'grassroots' concerns, assisted and facilitated by development experts. Thus the *participation* must involve an integration of local knowledge in the development process. Even if the post-development theories have meaningful sensibilities, they have been accused of not having a future programme, the lack of instrumentality. Nustad points that having a meaningful sensibility is a value in itself. In addition, the post-development approach should acknowledge that the apparatus in which development is embedded in itself has certain effects and is built on certain assumptions. In addition, it is important to study the manifestations of development in concrete encounters. (Nustad 2001)

Corbridge (1998) admits some strengths of post-development thoughts. The post-development thinkers force us to confront our own prejudices about the agendas of development and some examples of shocking failures. They also provide a human touch that is too often missing in development studies.

However, Corbridge (1998) argues that even if the discourse of development originated in the first flush of the cold war, and because the age of development has been associated with famine, debt, and the ravages of structural adjustment, we must think twice before we abandon development. He refers to some post-development writers, for example Esteva and Prakash (1998). They argue, in short and simplified here, that having stepped outside the diseased circles of modernity, science, reason, technology, westernisation, consumption, the nation-state, globalisation and development, the people of the social majority can then make and rule their own lives at grassroots. The key to good life is in simplicity, frugality, meeting basic needs from local soils, and sitting together in the commons. Corbridge argues that post-development fails to convince because it too often trades in dogma and assertion, and too rarely resorts to proper argumentation. He makes some objections to post-development:

1. Post-development makes its case for change with reference to unhelpful and essentialised accounts of the 'west' and the 'rest'. The west is inauthentic, urban, consumerist, monstrous, utilitarian, and the people are lonely, anxious, greedy and shallow. The emptiness of western lives is not without merit, but too neat and partial. (Corbridge 1998)

2. Post-development equates western science and reasons with technology, and bad technology at that, but science as technology can be liberating where it is labour-saving or life-saving. In addition, science can be liberating where it is lit up by a healthy scepticism or a bias for falsificationism. (Corbridge 1998)

3. Post-development activists fail to acknowledge that critical thinking and open political systems might be preconditions for the 'pluriverses' they wish to

celebrate at grassroots. (Corbridge 1998)

4. Post-development romanticises the 'soil cultures' of the social majorities. (Corbridge 1998)

5. Post-development fails to consider the opportunity costs of the spatial closure strategies it commends on behalf of the social majorities. For many, knowledge is power and the failure to make clear the costs of post-development does not reflect well on the moral landscapes of it. (Corbridge 1998)

6. Post-development fails to address the downside of delinking because it associates the ills of the world with development, but maybe debt, famine etc. are the results of an absence of development. (Corbridge 1998)

7. Post-development gives no hint of the extraordinary accomplishments that have defined the development, e.g. increased life expectancy of people. Development is about dilemmas, and the shortcomings should not be read as failures only. (Corbridge 1998)

Nederveen Pieterse (2000) points out that post-development thinking is uneven. Its use of language is sloppy and indulgent. Post-development is based on a paradox. While it is part of the broad critical stream in development, it shows no regard for the progressive potential and dialectics of modernity – for democratisation, soft-power technologies, and reflexivity.

I like to consider development as a change for the better. All in all, there must be some common sense involved in the ideologies of development or post-development. It is not an either-or question. Why not grassroots ideas which can be combined to development, keeping in mind a self-reliance and sustainability.

Ullrich (1992) argues that one important reason for the inequality and superiority of the North is the development of science and technology, as tools for progress. In the following, I would like to examine this question, when it comes to information technology and how it could be utilized in developing countries. By combining this question with the development theories above, I have tried to clarify my relation to these theories as an information systems scientist.

4.2 Information Technology in Africa

The development of information and communication technologies has been very fast over the past few decades. We can see applications of IT everywhere in our society: in industry, health care, education, agriculture, travel, and economic activities, just to mention a few. Thus a technological revolution, meaning mainly information technologies, is reshaping the society (c.f. Castells 1996). The social and cultural environment in many countries has usually had difficulties in following the speed of technological changes. When we consider the development of information technology, examined in Chapter 2, there is a huge information gap between industrialized countries and developing countries (e.g. Boer & Walbeek 1999). In western countries we are free to talk about the 'information age' or 'network society', but in developing countries access to information and communication technology is still very limited.

African countries, as well as the World Bank, United Nations, and many international agencies, not to mention researchers within developmental informatics (e.g. Woherem 1992, Korpela 1992, Odedra-Straub 1996), have embraced IT as a tool for development. This can include both socio-economic development and human development, which means to enlarge the range of people's choices and make development more participatory and democratic (Human Development Report 2001). Heeks (1999a) suggests that information and communication technology can play a role as communication technology, rather than as an information processing or production technology. The key message of the World Development Report 1998/99 for Africa (World Bank 1999a) is that most countries on the continent (Africa) need to do much more, much faster, to increase their knowledge base, to invest in educating their people, and to take advantage of the new technologies for acquiring and disseminating knowledge, and information is necessary for development (Kiangi & Tjipangandjara 1996). It is one way to promote democracy: "*Power is not money of few, but information in the hands of many*" (Olivier 1996, p. 183).

Thus the goals when implementing new technology are manifold. The implementation must go with minimal disruption, it must ensure acceptance from the end-users, and deliver the tangible benefits (e.g. improve the administration and management, productivity, efficiency) on time and within the planned costs (Lopez & Vilaseca 1996, also Woherem 1992). According to Waema (1996), the real test of economic development is whether it reaches people and whether it is sustainable. For developing countries, the jump to an information society can be a successful process, if the emphasis is not on technology itself, but in people. However, the general impression is about relative failure rather than relative success in the implementation of, especially large-scale, computer-based IS in developing countries (Walsham 1992). The operational effectiveness has been far below initial expectations, if not disappointing. The impacts of IT has been low. Thus the potential role of IT as a socio-economic developer has not been fulfilled (Odedra 1990, Odedra-Straub 1996, Waema 1996). Of course, there are other examples with the evidence that computers are used for critical operations successfully and the technology is highly valued (e.g. Oyomno 1996, Srivihok 1998).

Informatics is already causing changes in the ways in which government and companies work in a number of African countries. It is a powerful agent for technological and social change (Zwangobani 1990). As noted above, the expectations are not always met, and further, there can be some negative outcomes as well. Roche and Blaine (1996) examine some example of such outcomes. First, the introduction and use of IT can create a growing rift between the users and non-users in terms of education, location and poverty. This may lead to greater inequality and increased economic and social polarization. Second, advanced IT, especially in the manufacturing sector, can be linked to the growing unemployment in the developing countries, when related to traditional mass-production processes. The third problem concerns the cost of building and maintaining the basic infrastructure to support IT. Due to the shortages of basic needs, such as food, education, medical supplies, water, and sewage systems etc., certain parts of the world are in risk of being left out off the global information ‘super-highway’, hence becoming even more backward and isolated. They need to make substantial investments in roads, water, and sanitation, electrical production and telephony before they can hope to get the full advantages of IT. This problem is exacerbated by the large deficits and debts of many developing countries, also by their low rates of domestic savings and investments, and their limited access to international capital markets. Roche and Blaine (1996) conclude, technology itself is not a problem, but the problem is in IT uses and applications.

In the literature of IT in developing countries (main forums for such articles are a journal of Information Technology for Development, IFIP WG 9.4 conference, and recently also an Electronic Journal of Information Systems in Developing Countries EJISDC), the early articles focused on national strategies and policies, attempting to define the potential role of IT in developing countries. Later on the articles mostly concern an analysis or an examination of the technology transfer, cases of introduced IT systems in organizations or governmental institutions, or the diffusion of IT in different regions, or the effect or importance of existing or missing IT policy by the government, and lately the globalisation. Fewer articles can be found about ISD or software engineering in developing countries.

In the following I will briefly examine the history of computerization in Africa, and continue to specify what is the nature of the software industry in developing countries and Africa. This especially concerns issues like what is appropriate for Africa in terms of information technology and information systems, as well as how appropriate systems can be developed, what are the problems and strategies to solve these problems, in favour of local systems development.

4.2.1 Computerization in Africa

Since the 1950s computing facilities have been transferred to Africa and developing countries as a whole. First the flow was weak, but it has been increasing steadily. The first computers were large mainframe systems, usually inappropriate because of the size, capacity, power supply, effect on local labour etc., by the receiving organizations. It was not until the late 1970s that the computerization process started to increase in Africa and not until 1983-1984 (with microcomputers) that observers noticed a true informatics ‘boom’ in many African countries. Still in 1980, Black Africa was, along with the poorest regions of Asia, the least computerized area of the world. (Kluzer 1990)

Traditionally the emphasis has been on international aid or donor agencies’ efforts to technology transfer, meaning selecting and installing computers and computing systems developed elsewhere (e.g. Bell & Wood-Harper 1990, Kluzer 1990). Since a large portion of aid in Africa is spent on ‘basic needs’, like famine relief, elementary education or primary health care problems, using computers as part of a solution could have consequences that are hard to predict (Kluzer 1990). Many times these projects lack studies of new systems in terms of requirements and aptitudes of the resident user community (Bell & Wood-Harper 1990). Still in 1980s (Kluzer 1990) almost half of the African computer market is currently comprised of somebody's foreign aid. There are organizations such as the United Nations Educational, Scientific and Culture Organization (UNESCO), United Nations Industrial Development Organization (UNIDO), United Nations Development Program (UNDP), International Monetary Fund (IMF), Intergovernmental Bureau of Informatics (IBI), and International Bank for Reconstruction and Development (World Bank), which have had a critical role to play in promoting IT in developing countries. However, these institutions have been criticized for their excessive cost and bureaucracy, and general inappropriateness (Roche & Blaine 1996). The World Bank's policy has been criticised to be too techno-economic, ignoring socio-cultural aspects (c.f. modernisation) (Nulens 1998).

The introduction of SAP (structural adjustment programs) by the IMF and the World Bank, in 1979-1980, also influenced the computerization process. As a result, in general, a growing number of African countries conducted a far-reaching process of 'rationalization' of the state sector. This follows three major lines: cuts in government spending (hitting primarily public employment and investments), deregulation of agricultural markets (bans or radical changes in price fixation, state control on internal and external trade etc.), and deregulation of other service sectors like transport. Accordingly it means shutting down, restructuring, and privatisation of state owned enterprises. In general, the incorporation in a smaller range of state activities faced stricter planning and evaluating procedures, with emphasis put on financial viability and efficient use of resources. On the other, greater vitality and expansion of the private sector (indigenous and foreign capital), which, freed from excessive constraints, should begin to modernize under the pressure of increased competition. All this affected the increased computerization in Africa. (Kluzer 1990)

In 1980s there was a perceptible shift away from foreign aid and multilateral lending as the primary providers of infrastructural investments toward the use of private foreign and domestic capital. These market-based solutions lead to foreign borrowing and investments on the recipients' balance of payments, and social issues such as access to and pricing of essential services. Without going any further analysis of foreign capital (see Roche & Blaine 1996), Roche and Blaine conclude that foreign investment and international capital will not provide an acceptable long-term solution to investment in basic infrastructure and essential services in developing countries. The question seems not to be that simple. According to Heeks (1996b), many countries that seem to be moving from state to market in general terms, are simultaneously increasing the extent of state intervention.

The government, public administration and state-owned enterprises (SOE), and local branches of multinational corporations and foreign involvement have been the pillars of computerization in black Africa. Soon after independence, in the public administration sector, computers were acquired in order to face the sudden lack of adequately trained civil servants needed to run data-intensive procedures in areas like payrolls, treasury accounting, tax, and customs revenues management, national census, etc. Where foreign companies were allowed to work unrestrained, they also quickly installed computers responding to the need to conform to corporate procedures set by their headquarters on a worldwide basis, and later, with telecommunication development, for strengthening the central control and coordination of international operations. (Kluzer 1990)

In the 1990s, computers are still used for simple applications such as basic accounting and invoicing, or word processing, and their potential as instruments for analysis are often unexploited. Maybe the reason is that managers are not able to utilize computers for decision-making. The other reason concerns the inappropriateness of Western produced applications to the developing context. This calls for the need to develop applications more suitable for the circumstances of developing countries (Lind 1996).

Some numerical indicators. The world information technology market (including computer hardware, data communication equipment, computer software, and computer services) in 1995 was estimated US$527.9 billion as measured by the revenues of primary vendors (1.8 per cent of GDP). OECD countries accounted for almost 92 per cent of the market, and Asia, including Japan, shared over 20 per cent of the world market. Markets outside Asia and the OECD area remained at around 4 per cent of the world in total. Some Asian countries such as Singapore are at similar level with some OECD countries. There is also some great potential for growth in parts of Asia (e.g. China, India) and South America (e.g. Argentina). In these statistics by OECD, from African countries only South Africa was mentioned as a country, where the industrial or service sectors are sufficiently developed to require computer services. On the other hand, the relative share of *software* trade in Africa has increased from 0.1 to 1.7 per cent (including Kenya, Nigeria, and South Africa), while in North America it has decreased from 34.2. to 30.7 per cent. (OECD 1997)

The year 2001 Human Development Report by UNDP introduces a new measure of the technology achievement index (TAI), which indicates the countries' ability to participate in the network age. The TAI index "aims to capture how well a country is creating and diffusing technology and building a human skill base – reflecting capacity to participate in the technological innovations of the network age" (UNDP 2001, p. 46). The TAI is intended to help policy-makers define technology strategies. According to the TAI values, the countries can be divided into four groups: leaders (TAI above 0.5), potential leaders (0.35 – 0.49), dynamic adopters (0.20 – 0.34) or marginalised (below 0.2). Even if TAI measures technological achievements, it correlates highly with human development index (HDI), better with the HDI than with income.

The differences between some industrialised and developing countries in the penetration of communications equipment and the technology achievement are shown in TABLE 11. The numbers tell about the gap between the countries in the accessibility to the global information infrastructure. In the world, 80 % of the world's population has no access to reliable telecommunications, and one third has no access to electricity (Heeks 1999a; Panos 1998). According to UNDP

(1997) more than half of humanity in the world has never made a telephone call.

TABLE 11 Technology achievement indicators (adapted from World Bank 1999a, 2000, 2002, UNDP 2001).

	Nigeria	Ghana	Tanzania	South Africa	Finland	USA
TAI value	..	0.139	0.080	0.340	0.744	0.733
Telephones per 1000 people (1999)	.. (4 in 1997)	8	5	138	552	682
Mobile subscribers per 1000 people (1999)	..	4	2	132	651	312
PCs per 1000 people (1997)	5.1	1.6	1.6	31.6	310.7	406.7
Internet hosts per 1000 people (2000)	.	.	.	8.4	200.0	179.1

The total number of computers permanently connected to the Internet in Africa (excluding those in South Africa) broke the 10,000 mark at the beginning of 1999. As measured by Network Wizards, growth is up 36% in six months. The correct figure may actually be closer to 12,000 or 15,000 due to the measurement technique which cannot count hosts which are not properly reverse referenced in domain name servers. In any case this represents about one host per 75,000 people, or 0.024% of the world's 43 million hosts. Nevertheless, the six-monthly African host growth rate almost doubled the world average (18%), so this a significant increase on the earlier figure of 0.021% in July 1998. Yet all of Africa has about as many Internet sites as Latvia, with a population of 2.5 million. (Jensen 1999)

4.2.2 Software industry in Africa

Dependence on foreign skills, technologies and products has characterized and will continue to characterize the evolution of the computer sector in most African countries (Kluzer 1990). Most of the hardware, software, and technical knowledge has been imported (Avgerou 1996). However, the real *technology transfer* should refer to the situation when a particular technology can be developed, repaired, managed and operated domestically. The country "has acquired not only the operational knowledge but also the deep technical know-how and tacit knowledge of the given technology" (Woherem 1992, p. 73). Korpela (1994) refers for example to Wisner (e.g. Wisner 1990) when criticizing the transfer of technology. Wisner presents 'anthropo-technology', where in the technology transfer, the success or failure is related to how the *totality* of the technology is transferred. This totality includes organizational and maintenance aspects, the supporting network, the existence of which we are so used to that we do not see it. The question is about transferring knowledge and know-how (Neko 1992). In addition, the implemented technology often triggers changes in the structure or practices of the organization, for example new requirements for management and sustained use of resulting IS (Avgerou 1996). According to

Avgerou, when transferring practices, organizational structures, routines, and skills that have proved valid in a different context, one should be extra cautious. Accordingly, similar caution should be exercised when adopting IS development methods or methodologies. Western methods and methodologies are not necessary applicable within an organisational culture in developing countries (Madon 1994, Avgerou 1996).

Hence, when talking about IT or IS in developing countries, why do we assume that the question is only about the *transfer* of technology (in a broad meaning), which in many times is about physical dumping of technological goods made elsewhere (Woherem 1992). The question of whether the pure transfer of technology from industrial countries to developing countries is too concentrated on technology, leaving the social context without notice, has given a further question of *appropriate technology* (e.g. Avgerou & Land 1992, van Ryckeghem 1992), and how it can be achieved. I am coming to that question a little bit later in this section.

All in all, instead of speculating about the transfer of technology or *ISD for* developing countries, why not consider the *ISD in* developing countries. Since the advent of microcomputers, the birth of small IT companies in many African countries has taken place, usually starting as dealers or distributors for foreign computer firms. On this basis, also more sophisticated activities have emerged for example to hardware maintenance and hardware production, but also to consultancy and software services. (Kluzer 1990)

In the 1980s most efforts of IT companies were concentrated in the consultancy and computer services business. According to Kluzer (1990), in the 1980s just a few cases were reported of application software packages developed by African countries (in Zimbabwe, Cameroon and Ivory Coast) for the local market and /or sold abroad. As pointed out earlier in this chapter, the intended outcome of IT depends largely on the ability to develop IT applications to meet local circumstances (Avgerou & Land 1992).

Today the creation of a local software industry in developing countries, indeed in Africa also, is regarded as essential in order to adapt software technology to suit particular local needs (Heeks 1996a, Heeks 1996b, Hassan 1998). In addition, software production has low entry barriers since it is less capital-intensive compared to hardware production, it is labor-intensive, and has a lower rate of obsolescence (Heeks 1996b, 1999b). Heeks (1996b) approaches this local ISD from a software industry perspective. Surely the global software market with global software outsourcing and India's experience (e.g. Bhatnagar 1994, Nicholson et al. 2000) is interesting and challenging, but Grundey and Heeks (1998) also discuss, what they call '*natural protection*'. It means protection of local software development based on unique user requirements, derived from particular organizational practices and social, economic, political, and cultural environment. The elements of natural protection include the spoken language in the country, the importance of personal contacts rather than mass marketing, and the process of transition and the continuous changes it demands. Protection also derives from the high cost of imports. These protective factors influence that local software consumption is delinked from the global marketplace, to some degree. In this thesis, the interest is more in ISD as a work practice, and what the problems are and opportunities.

Appropriate information systems. Modern information technology has been developed to serve the needs of industrial countries for economic growth. This technology is complex and expensive, hence many have doubts about the appropriateness of IT for poor countries (see Avgerou & Land 1992). According to Avgerou and Land (1992), most development programmes are based on economic and technical considerations, even if there is a need to include social criteria when devising technical solution to a given situation. In addition, as noticed in the chapter on Information Systems, computers are only a part of the information system, which is always complemented by manual procedures as well as informal ways of communication and action, also in developing countries (c.f. Avgerou 1990). Thus even if the introduction of IT in developing countries has been a disappointment in many cases, it does not mean inappropriateness of computer technology as such, but a lack of adaptation to the local society (e.g. Korpela 1990).

Avgerou and Land (1992) review the literature of development studies concerning appropriate technology. They find two different interpretations of appropriate technology. The first, and the most widely held view, is an *economic growth through industrialization*. Accordingly, appropriate technology means a strategy for economic growth (which has been noticed many times in this thesis), supported also by e.g. the World Bank and the United Nations agencies. It is an approach to select or develop systems, which fit the socio-economic context and can lead to further technological, economic and social development. According to authors, within this approach, appropriate technology is an approach of technology transfer. The other view gives priority to the satisfaction of *basic human needs*. It is an ideology that suggests a development, which focuses on empowering small local communities to be self-reliant. Appropriate technology should support this ideology, which aims at self-reliance, self-sufficient social units, human-enhancing labour, harmony with nature, and cultural autonomy. This should be considered as a preferable objective of development through traditional technologies (Avgerou & Land 1992).

However preferable the objective the ideology of development of IT as empowering humans to improve their living conditions is, Avgerou and Land (1992) consider it as utopian. If a community acquires the capacity to manage IT within its activities, it has already taken a very significant step towards modernization. On the other hand, if the community does not develop such capacity, the technology is usually produced, marketed, and supported by the multinationals of the industrialized countries, which is not the goal of self-reliance. For the other perspective of socio-economic growth, where the emphasis is on the need to build the capacity of responsiveness to local developmental needs and constraints, the question of appropriate technology is meaningless. Avgerou and Land (1992) examine the strategies to be adopted to lead to appropriate IT systems in this sense, including some policy making strategies by the government. In ISD, the authors emphasise the characteristic of systems development as a process of socio-technical change. So, systems developers should consider the socio-economic appropriateness of the technical options during critical tasks of the process, such as the planning of future major information systems changes, requirement analysis, feasibility study, and implementation. However, most methodologies have little methodical effort to match technical and socio-organizational change. Thus the contributions to improve the organization within which IT is embedded, very often fails, particularly in developing countries. (Avgerou & Land 1992)

Avgerou and Land (1992) suggest some issues as preconditions that should be met for effective implementation of IT. Even if these preconditions are taken into account, it does not guarantee the development of appropriate systems. However, it can help to avoid the waste of attempting to develop ineffective systems in developing countries. The first precondition is *management competence* and managerial understanding. The lack of highly trained managers in both private and public sectors is a common situation in developing countries. The other critical precondition concerns *shared values* within an organization. Shared values should include values of the work force at all levels. In developing countries this can be difficult, since the use of ex-patriates at senior level can create contradictions with indigenous staff, when it comes to shared values. In addition, the traditional hierarchical structures can be deeply embedded, and if this is threatened by new IT, difficulties can be expected. The last precondition concerns *organizational changes*. The introduction of IT many times creates changes in an organization, and the change can run across deeply ingrained organizational norms and structures. In developing countries such norms and structures are indeed robust, which can lead to the failure of implementation. Besides these preconditions, the authors emphasise adequate *channels of communication* between work force and management, so that all are aware of what change is planned. In addition, a high level of *education and training* of the work force is essential. Well educated people can better understand the nature of the changes, adapt the changes and adjust their skills to new demands. Thus it is important for an organization to provide continuous education and training for its staff. (Avgerou & Land 1992)

Also Korpela (1992) suggests a couple of preconditions, especially regarding appropriate health informatics. Firstly, the focus must be on *cooperative systems development* for priority needs, not just on hardware imports. The second one he mentions is a *network of supporting activities* needed. I would add here the *user empowerment* (c.f. Bishop 2000), related to cooperation. The user empowerment means the participation in all stages of the process.

Korpela (1994) criticizes Avgerou and Land's (1992) analysis by two theoretical deficiencies. First, the authors only discuss the British socio-technical tradition by Checkland and Mumford, which is based on an assumption of shared interests and consensus within an organization. Korpela compares it to the critical Scandinavian tradition, and points that the 'appropriateness' or 'success' of IS can be different from the different perspectives of different stakeholders. Secondly, Korpela criticizes their approach to the question of developing countries, which is based on traditional – modern dichotomy. He reminds that Britain is often regarded as one of the most rigidly class-divided societies in Europe. However, Avgerou and Land take the appropriateness discussion to concrete level. In addition, they ask for the methodologies, which assist organizations in developing systems that are 'appropriate' for the organization's end. (Korpela 1994).

Avgerou and Land (1992) consider the other above mentioned ideology of the development of empowering small local communities to be self-reliant with the help by IT, to be utopian. This kind of application can be for example a health care system or an educational system. They argue that if the community reaches the capacity to plan, develop, implement, and maintain any IT, they take a step toward modernization. Maybe yes, but why not toward development? Yes, this step needs new kinds of resources and facilities, and changes in their traditional activities. In extreme cases, the users of IT adopt a view of the world that is fundamentally different compared to the basic cultural values of others, thus creating a growing rift within the society (Roche & Blaine 1996). In all societies technology (any technology) has always been subjected to change (Andah 1992). People have devised tools for improving health, raising productivity, and facilitating learning and communication. Why then is IT such an either-or issue? The need for some other resources, facilities and changes in their activities, compared to traditional ones, does not exclude the appropriateness of IT at once. Indeed, what does 'traditional' mean, anyway? Does tradition change over time, or not, cumulatively? I think that there is a lot of variance between the 'traditional' community, and an image like "the electronic cottage is a dream of rural Californians, not of rural Africans" (Avgerou & Land 1992, p. 31). After all, information technology is just a tool, a new kind of an artefact.

Concerning computers, the need for imported hardware decreases self-reliance, but is that not the case everywhere (c.f. Korpela 1992). At this point, I also refer to Chapter 2 and to sustainable technology. One meaning of sustainable technology is that the technology should be able to be *used* without dependence on external assistant (c.f. Oyomno 1996). Oyomno also suggests that IT and IS can strengthen the management capacities, for example in public institutions, and thereby provide a rational basis for the attainment of sustainable development.

Pellegrini (1980) view of the desirable role of technology in development is germane to this discussion:

> *Technology should be considered 'appropriate' when its introduction into a community creates a self-reinforcing process internal to the same community, which supports the growth of the local activities and the development of indigenous capabilities as decided by the community itself (Pellegrini 1980, p. 1).*

The penetration of IT systems is still quite low in Africa, there is ill-utilisation and under-utilisation of systems. Most of the systems are designed by consultants from abroad, or brought through off-the-self systems from vendors, or they are pirated. The problems when using IT in Africa (like in many other developing countries) can be grouped into *operational problems* (like technical constraints, lack of skilled staff, under-utilisation, inability to maintain the system), *contextual problems* (like inappropriateness of Western produced system, semantic differences between phenomena is understood and how things are worded, differences in value systems, differences in what is acceptable or reasonable), and *strategic problems* (like the diffusion of IT and how scarce resources are utilised, the selection of appropriate applications with particular emphasis on developing problems). (Woherem 1992)

All in all, many systems fail in Africa as a result of the cultural mismatch between the Western produced system and the recipient African culture. IT systems, like any other technology, are not culturally neutral (Woherem 1992), or should I say, contextually neutral. That is why we consider it important that critical IT applications are developed locally, or at least adjusted locally, if the origin of the application is from abroad.

ISD in developing countries – related research. Many developing countries are nowadays part of the global software business, mainly because they have cheap and talented labour, relatively low entry barriers and few economies of scale (Heeks 1999b). A good example is India, being an example of a country applying export-oriented strategy. However, instead of software packages, the main bulk has been in software services. In addition, the question is mostly in low-skill software construction and testing, leaving the high-skill development in western experts, but, according to Heeks (1999b), the Indian success makes many developing countries follow in its footsteps.

Besides export-oriented software business, there is an option of domestic market. There are two options for that: package development or software services (Heeks 1999b). According to Heeks, providing packages for the local market in the general application market of word processing software, spreadsheet, databases, operating systems, and the like, is not a profitable option. Imported packages create the entry barriers for local firms formidable, including pirated packages. That is why the majority of software companies in developing countries provide software services for local companies, but as Heeks points out, the domestic software market is not very sizeable, or even demanding. However, Heeks sees it as a good starting point for further progress, also into exports, even if the question is more on survival than development (of the company). Heeks (1999b) also names one additional option for software business, focusing on niche markets, such as sectoral niches, application niches or linguistic niches.

In this study, even if these strategic evaluations are interesting in general and important for the company, we concentrate more on the basic questions of systems development. What comes to the related research concerning my research question of ISD practices in developing countries, more specifically in Africa, there are not too many studies of that. In the following I present a few examples of systems development projects described in the literature of development studies.

Macias-Chapula et al. (1998) describe a system development project at the Mexican Institute of Petroleum (the intelligence system to support the innovation processes at the company). In the project, a settled working group used the soft systems approach (Checkland 1981) to analyse the situation, i.e.

problem situation regarding the access and use of information. At the end, the authors summarise some challenges for the future of the project, e.g. the need to integrate both soft and hard approaches in the development of the technology information (intelligence) system, the need to foster a culture of problem situation understanding as a necessary prelude to further work towards information systems development, the need to incorporate new elements and accommodate changes within organisational culture, the need to promote the socio-technical nature of the system thus involving the personnel in decision making, and the need to obtain performance indicators of the system, including soft elements such as the value of the system at the individual and institutional/organisational level.

Lai (1996) reports a case of the development of an IS for a procurement office in the Philippines. She applied Checkland's (e.g. 1981) Soft Systems Methodology for systems enquiry and Strategic Data Planning Methodologies (Martin 1982) for data analysis, in action research manner in the company. She concludes that complete solutions can be found when different methodologies from different disciplines are employed in information systems development. Nicholson et al. (2000) describe a longitudinal case study of the process of managing IS development in a UK company's subsidiary in Bangalore, India. They applied Giddens' (1984) structuration theory and his later writings on globalisation (Giddens 1990), and approach the ISD from the UK managers' viewpoint. They conclude for example that the company needs to adjust their management approach to the local (Indian) context.

Most ISD studies in developing countries still focuses on the target system (the product) and the success of it. In general we can summarise that the success of new IS is dependent on the proper combination of 'technology transfer' and local adjustment. The local socio-economic context is considered important. However, empirical data about how these systems are developed or maintained in practice is missing: what are the ISD practices in developing countries, and especially in Africa. The potential contribution of the technology transfer to local systems development can be assessed only if we understand the current practices and needs of the local software companies (c.f. Duncombe & Heeks 1999a).

Korpela (1994, 1996) studies systems development activity and network in a case of a Medical Records system of the Obafemi Awolowo University Teaching Hospitals Complex in Nigeria, using the activity theoretical framework. I will come back to that case in the following, when discussing the Nigerian situation in an African context.

Constraints of indigenous systems development. In case of ISD, the paradigm of ISs as social systems implies the need to understand social factors and, more generally, the organizational and national context within which information systems are to be implemented (Walsham et al. 1990). However, many countries find it problematic to integrate new technologies into existing institutional structures (Roche & Blaine 1996, Waema 1996). Many times, for example in Africa, if a computerization project has not failed, the delivered technical system does not have a significant positive impact on the performance of the organization or the equipment is under-utilized (Odedra 1992). There is extensive literature on the problems within information technology and systems development. Waema (1996) gives a good summary of the main issues which we have supplemented in the following (see Korpela et al. 2000b, Mursu et al. 2000):

1. Many problems relate to *inadequate infrastructure* (see also Bhatnagar 1992, Woherem 1992, Roche & Blaine 1996). The most noticeable are poor power supply and telecommunications. Due to the poor power supply, it is necessary for example to equip all systems with an Uninterruptible Power Supply (UPS) to protect equipment against voltage variation and power failures (Okot-Uma 1992). Also, private businesses use quite much money to guarantee the power supply: about 90 percent of firms in Nigeria had installed a generating capacity (World Bank 1994). The poor telecommunications utility has resulted from a lag in the development of the national data communications network.

2. Another group of problems concern the *shortage of skilled personnel* (see also Odedra 1992), leading to *operational problems* (Okot-Uma 1992). This problem relates to *poor maintenance* (Mundy 1996) and *lack of planning* or *poor planning* (Woherem 1992), and *inability to manage change*. Many times these are management problems, and concentration is on the technical aspects of the system, instead of socio-organizational and business factors (Woherem 1992). IS professionals and managers are not educated in managing complex ISD processes. The education emphasis is on software engineering instead of information systems. Thus system developers in Africa work under severe practical constraints but are less adequately trained to cope with them than their colleagues in the industrialised countries.

In addition, *new technology* and *shortened product life cycles* (Lopez & Vilaseca 1996) can cause complex situations. Rapid technological change requires resources to research, develop, manufacture, and support technology. New technologies also demand managerial and organisational changes in order to receive the maximum benefit from their application. On the other hand, the problem can be in fast technological obsolescence.

3. The third group of problems relates to the *unsupportive public sector culture* (Waema 1996), which can be seen as a *colonially inherited administrative culture* (Korpela 1996). Ojo (1992) names some 'socio-cultural' and organisational issues influencing the development and outcome of IT applications for example in Nigeria. These issues comprise of an *over-politicised decision-making process*, *bureaucratic complexity, commitment to personal gains, culture of secrecy*, and *preference for informality*. In addition, the history of the software industry is young, so most vendors of IT systems have been with a business mind-set "dump the goods and run" (Woherem 1992, p. 74). This has not improved the reliability of local software vendors.

Waema (1996, also Okot-Uma 1992) also emphasise *the lack of national IT policy*. A national IT policy would give guidelines on how information technology would contribute to the social, economic, and political conditions in a country. Also the *lack of IT awareness* in the society and also on the part of business managers and government officials is remarkable (Woherem 1992).

4. Many African countries have grave economic and political problems which cause *insecurity of life and uncertainty for the future*. This is a formidable hindrance to long-term initiatives like ISD.

Strategies for acquisition of IT and indigenous ISD. African countries must carefully analyse the potential economic, social, and political costs of introducing IT and develop specific policies to adapt these technologies to their local context and needs (Woherem 1992, Roche & Blaine 1996). This is also a starting point for sustainable utilisation of IT. The possible leap frogging of technological development is dependent on the current level of technological capacity and other resources of each country. The ability to develop one's own IT industry is dependent on the size, level of educational facilities /sophistication, level of industrialisation, gross national income, per capita income, population, etc. (Woherem 1992).

According to Heeks (1999b), the three main efforts of those countries that have succeeded in creating local software business are the following: 1. *Enterprise tactics*. This includes e.g. effective mechanisms for information and knowledge transfer in formal and informal networking, identification of demand-growth markets and synergies, cost or service innovation and good marketing. This also includes sufficient access to investment and working capital. 2. *National strategy*. Governments should act as industry promoters. This includes financial actions (e.g. by enabling private sector and overseas funding), investments in education and training, as well as research and development, and taking care of intellectual property rights (e.g. piracy). Last but not least is the infrastructure. 3. *National vision*. All plans and efforts emerge and sustain only if there is a national vision. Thus developing countries should learn to combine successful tactics, strategy and vision.

Waema (1996) suggests strategies to successful implementation of IT projects on three levels of economic development: national, organizational, and individual or group levels. The national level includes strategies such as providing national IT policy by the government (see above), which creates an enabling environment (including infrastructure). Waema suggests that it can encourage the evolution of a local software industry capable of developing software applications for local needs and local market, but also for the international market. The IT policy should be included in the national development plan. The other national level strategy is cultural re-orientation by government, meaning a culture of IT use in government and business. The desired changes include democratic structures, strategic thinking, and less political power dominance. The organizational level includes strategies such as effective project leadership to manage the change process and empowering users to control their activities by education, training, and participation. Organizations should also learn by experimentation and piloting. On the individual or group level the strategies could be lobbying, trying to influence IT policy, and value system change. This means positive attitudes, commitment, and so forth.

The complexity of the socio-cultural and organizational context within which IT applications take place in many DCs, requires IT experts with appropriate knowledge and skills to deal with that context of social, economic, organizational, and cultural conditions. User involvement is a major issue that brings a social dimension into IT application. One social dimension is people's institutionalised work culture that the IT application will probably change. IT application imposes strict work discipline on their users which could require them to do away with their traditional work culture and to imbibe the new IT culture. (Ojo 1992)

We agree that it is particularly important for Africa to include broader organizational, social, and political issues into systems development, and that educational and other 'empowering' strategies are particularly important (Walsham 1992, Waema 1996). The technology of information systems is likely to be the same in industrialized and developing countries, but the uses and preconditions differ. Thus ISD in different countries needs to be adjusted to take the infrastructural, organizational, social and political differences into account.

4.3 Nigeria – a country of opportunities?

In the following, Nigeria as a research context is presented to provide background information for the study. The history is illustrated briefly, followed by the description of the IT situation in the country. Part of the country description, the results of the survey study of 'a typical software company in Nigeria', conducted within the INDEHELA-Methods project, are also presented.

4.3.1 Basic facts of Nigeria – history and now

It is not the purpose here to present the Nigerian history throughout, but it is reasonable to give a brief summary related to African history above, in order to illustrate the development of present-day Nigeria. Nigerian history is part of the western-central African history. The first evidences of indigenous groups are from the Stone Age around 12,000 BC (e.g. in Ile-Ife, the home town of OAU). One of the most famous figures found in Nigeria relates to the introduction of iron smelting and the widespread Nok culture between the fifth and third centuries BC. During 1000 – 1500 A.D., there was a foundation of many kingdoms such as Benin, Oyo, Hausa states, and Kanem-Borno. The growth of the states was often based on slavery. The first contacts with Europe and the New World took place in 1450 – 1850 A.D. during the slave trade. At the turn of the nineteenth century, an Islamic revolution took place in northern Nigeria. The spread of Christianity enlarged in the nineteenth century, mainly by African missionaries. Christianity and Islam became the two dominant religions. Pre-colonial societies relied on agriculture (e.g. yam, cereal, groundnut, oil palm), manufacturing (e.g. iron working), and the trade of high quality ceramics, wood, textiles, and leather. (Isichei 1983, Falola 1999)

The European conquest of Nigeria began in 1861, when a British consulate was established in Lagos. A single firm, The Royal Niger Company, received a charter to govern the trade, a charter which lasted from 1886 to the end of 1899. The Royal Niger Company had a dramatic impact of the life of a small number of states, which were situated on the Niger or its tributaries, but outside these areas, life continued on as before. In 1877, a prolonged war was started among the Yoruba in the southwest, which was negotiated to end in 1886. British Protectorates were established at the end of the nineteenth century. The name 'Nigeria' was officially adopted in 1897. (Isichei 1983, Falola 1999)

Nigeria as a modern political entity was created in 1914 by the British. The ethnic groups did not develop into a strong nation during the colonial period, indeed, a policy of divide and rule kept them apart. The colonial economy relied mainly on exports, a trend that continues till today. Cash crops areas benefited immensely. However, in the past, Nigeria was a major food exporter, today she is a massive food importer. This has been caused by the expanding population and urbanisation. (Isichei 1983, Falola 1999)

During decolonisation, there emerged some influential parties. The NCNC under Nnamdi Azikiwe took a lead in southern Nigeria. In 1959 a new party emerged in the Western Region, the Action Group (AG) under Obafemi Awolowo. In the northern region, the Northern Elements Progressive Union (NEPU) came out under Aminu Kano. From the Northern Region developed a far more influential party, the Northern People's Congress (NPC) under Sir Ahmadu Bello. These parties and their leaders created a strong nationalist pressure. (Davison 1983)

Nigeria achieved her independence from Britain in 1960. The country was dominated by three large groups, Hausa-Fulani, Yoruba, and Igbo. The three regions and their ethnicities competed as enemies, which led to civil war that lasted from 1967 to 1970. The war consumed over a million lives, mostly on the Biafran side. The country became a parliamentary democracy at independence but has been under military rule from 1966 to 1979 and from 1983 to 1999. The army plays a prominent role in society, not for its ability to fight, but for its capacity to consume resources and prevent the growth of democratic politics. There has been inter- and intra-ethnic violence, including the above mentioned civil war. Nigeria is a multi-ethnic nation with more than 250 languages, English being the official language. Conflicts between Muslims and Christians have emerged since 1980s. About 50 per cent of the population are Muslims, concentrated in the north and southwest, about 40 per cent are Christians in the south and Middle Belt, the rest practice an indigenous religion. (Falola 1999, World Bank 1999b)

Nigeria is a huge country bordering the Atlantic (FIGURE 34). The neighbours to the west are Benin and Togo, to the east Cameroon, and to the north Niger and Chad. It is three times the size of the United Kingdom. Nigeria is the most populous country in Africa, representing about 20 per cent of the entire Sub-Saharan African population. With a current population of over 120 million people, Nigeria continues to record an annual growth rate of 2.83 per cent. The country has 36 administrative states, and the Federal Capital Territory. The federal capital city has been Abuja since 1991. The old capital, Lagos, remains the main hub of diplomatic, media, commercial and banking activities. (Falola 1999)

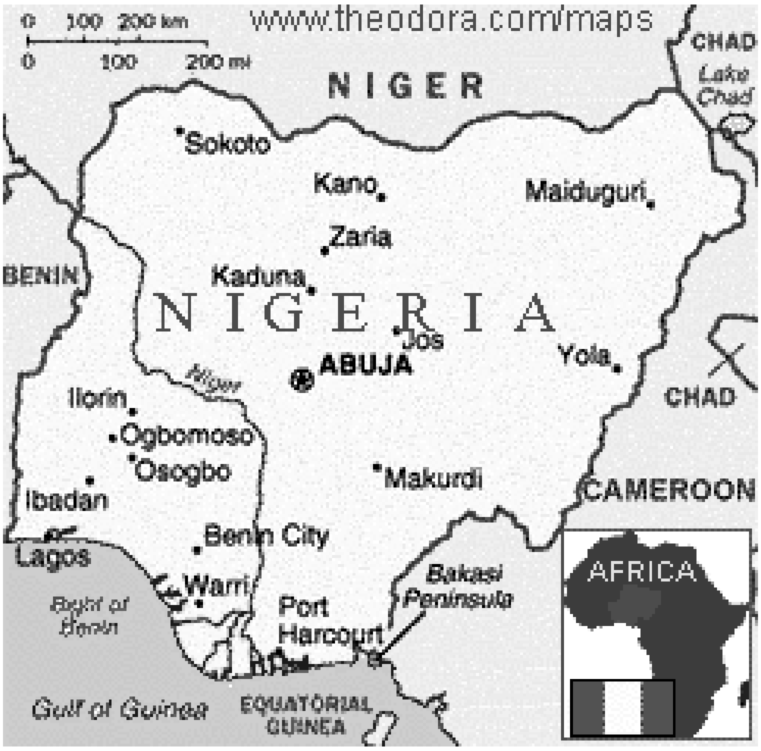


FIGURE 34 The map of Nigeria (http://www.theodora.com/maps/new/nigeria.gif)

The climate ranges from tropical with swamps and rain forests in the south, to arid with the savannah and semi-desert in the north. There are two distinct seasons, wet and dry. According to the climate, there are two vegetation types: the forest in the south and the savannah in the north. Settlement patterns and cultures are partially affected by vegetation, fishing and farming in the coastal zone, and agriculture in the forest zone. In the savannah zone, livestock rearing is combined with farming. In the rain forest, keeping cattle and horses is difficult because of the tsetse flies. (Falola 1999)

The geology is rich with tin, uranium, columbite, coal, gold, lead, iron, salt, copper, and zinc. From an economical point of view, a major feature since the 1970s has been Nigeria's dependence on crude oil (over 90 % of total exports). Industrial activities revolve around mining, manufacturing activities include textile production, beer brewing, and building materials and cement. Agriculture is the mainstay of the peasantry (cotton, cocoa, rubber, cassava, millet, maize, rice, and yams). (Falola 1999, World Bank 1999b)

Urbanization has been growing throughout the twentieth century, 36 per cent live in urban centres. Education and health care is quite developed, imported from western patterns. The adult literacy rate is over 51 per cent. Still, the life expectancy at birth is fifty-eight years for female and fifty-four for male. Major diseases include malaria, river blindness, meningitis, and AIDS. A number of people continue to consult traditional healers. The country is experiencing a series of environmental problems, for example the encroachment of the Sahara desert and soil erosion, slums in cities, oil pollution, and the illegal dumping of toxic waste from Europe along the sea coast. (Falola 1999)

Economically, before the oil boom, the growth was rapid until 1973, when oil-price increases provided unimagined wealth. Each region had a valuable cash crop, there were strong commercial classes, and government fostered local capitalism. Industries depended mainly on imports. Between 1968 and 1977 government revenue multiplied thirty-four times. Yet, in a bitter irony, growth of per capita GDP then slowed to 1.7 per cent a year during the 1970s and a decline of 1.1. per cent a year during the 1980s. (Okuwoga 1990, Iliffe 1995)

During the oil boom period, which lasted till 1981, Nigeria's economy was allowed to be heavily dependent on a single source of foreign exchange earnings. Oil was an enclave with only financial linkages to the rest of the economy. Its earnings overvalued Nigeria's currency, so that cash-crop exports collapsed while cheap manufactured imports undercut local industry. Suddenly, the boom period ended and Nigeria faced the problem of foreign exchange. The international depression of 1979-83 and a decline in oil prices in 1983 almost halved public revenue, created a foreign exchange crisis, boosted public borrowing, and inflation, reduced industrial capacity utilization below 40 per cent, and threw the economy into disorder which still reigned a decade later. External borrowing increased, there was accumulation of arrears, inflation and unemployment. (Okuwoga 1990, Iliffe 1995)

In order to normalise the situation the Government of Nigeria embarked on a Structural Adjustment Programme (SAP), aimed at restructuring and diversifying the productive base of the economy so as to reduce the dependence on oil and on imports. The policy instruments of SAP included for example a deregulation of foreign exchange, the lifting of import and export restrictions, and the privatisation of erstwhile government-owned companies and parastatals (UNIDO 1989). A number of measures were taken (Okuwoga 1990, p. 107):

1. introduction of a second-tier foreign exchange market
2. adjustment of tariff to favour domestically produced goods
3. import substitution of essential commodities from foreign countries etc.

As a result of these measures, industries found themselves in a tight corner and the task of management became very complex. Some companies faded away and those that remained in business came to realise the in-evitable need for computers as a means of providing a complex, accurate, and timely information for effective planning and control in order to survive. (UNIDO 1989, Okuwoga 1990)

The economic and political programs of SAP failed during the order of General Babangida in 1985 – 1992. In 1993, there was a presidential election, won by Chief Abiola, who received a clear majority of votes across the ethnic, regional, and religious divisions. However, Babangida annulled the election. Since 1993 the scene has been depressing. In 1994 – 1998 the power was in the hands of the military junta and General Sani Abacha, who died in 1998. So did the imprisoned president-elect Abiola. The Abacha regime lacked credibility both at home and abroad. The standard of living in Nigeria decreased so that Nigerians are among the poorest people in the world. Nowadays Nigeria is trying to turn into a democracy under president Olusegun Obasanjo, who was elected in 1999.

According to the World Bank Country Data (World Bank 1999b) the overall economic growth in Nigeria has been very low in the past few years, despite economic stabilization policies that have reduced average inflation from over 70 per cent (1994) to under 10 per cent (1998). The main factors contributing to the poor economic performance have been political uncertainty, poor governance, corruption, and inefficient state-owned firms in the oil, electricity, and telecommunication sectors, leading for example to the erratic electric power supply and poor petrol availability in the country. The structure of the economy in 1997 was (in % of GDP) agriculture 32.7%, industry 46.9%, and services 20.4%.

4.3.2 IT in Nigeria

The first computer in Nigeria was introduced in 1962 (Kluzer 1990). In the early sixties, computer-based ISs started with some third generation computers. The technology was new, the scope was limited and the available pool of technical knowledge was insufficient. Even though some people and organizations did realise the importance of the computer as a means of obtaining speedy, accurate, and relevant information, the level of awareness was too low as to have any significant impact on the social and economic development of the country. (Okuwoga 1990)

At that time, the pre-oil boom era, the sources of foreign-exchange earning were mainly from cash crops and the economy was relatively stable. Not much emphasis was placed on information processing especially as industries were seemingly being run effectively, there was far less need for crisis management which necessitates dynamic flow of information for quick decision-making at all levels. People could not therefore see enough justification processing for computer-driven ISs. By the early 70s, the first sign of meaningful steps towards computerised data processing was noticeable in government and some big industrial concerns. (Okuwoga 1990)

In the field of education, there was a rapid expansion of tertiary institutions and in the range of courses available at the institutions. Computer science departments started to appear at universities and polytechnics from 1977, offering degree-level education in computers and related subjects (UNIDO 1989), for example at the university of Ile-Ife (later Obefemi Awolowo University).

The bulk of the computer installations are in the Lagos metropolis. According to Okuwoga (1990), the first users in Nigeria include government (Federal Office of Statistics since 1963, The Federal Ministry of Finance since 1978), government parastatals (West African Examination Council since the 1960s, Nigerian Ports Authority, The National Electric Power Authority, Department of Customs and Excise, The Nigerian Armed Forces), government-owned educational institutions (universities, polytechnics and research institutes), and private sector (finance houses and manufacturing industry). Some of the applications were in-house developed, but usually from abroad. In the industry the first computers were adopted for example in such companies as the Nigerian Textile Mills (1970), Nigerian Agip Oil Co. (1971), Shell Petroleum Development Co. (1972), and Volkswagen of Nigeria (1976) (UNIDO 1989).

According to the United Nations (1973), in the beginning of the 1970s the total number of computers in Nigeria was about 30 (in the USA the number was around 80.000). In 1980 the number was estimated at over 400 and in year 1996 4.1 per 1000 people. In 1981-86, Nigeria alone counted for 20% of all computer imports in Black Africa (Kluzer 1990).

Till 1977, there were three computer vendors in Nigeria, which were the local subsidiaries of overseas computer manufacturers dealing entirely with mainframes and minis. In 1988, besides the in-house departments, more than 200 registered companies offered computer-related services (UNIDO 1989). In 1994 there were more than 500 registered computer companies and their activities centred mainly on computer sales and maintenance services, software development and information building, computer education and training, word processing and bureau services, and marketing and economic research, amongst others (Alabi 1994).

In the last two decades, the application of IT has become a matter of priority in many public and private organizations in Nigeria. However, most of the applications are limited to routine clerical activities, instead of providing powerful information systems for efficient decision-making. (Ojo 1992)

Problems. It seems that the problems concerning the introduction and effective use of IT in Nigeria have not changed very much during the years (also compared to the problems in developing countries in general). One of the problems has been the lack of computer policy in the country by the government. Key arms of the government in Nigeria have issued various policy statements on the use and development of IT, notably computer education, but there has not been comprehensive, official informatics policy promulgated by the government (UNIDO 1989, Jaiyesimi 1990, Okuwoga 1990, Ojo 1992, Alabi 1994), until year 2001. There was a promise of such a policy in the opening speech of a national software exhibition by the government representative in spring 2001, and such a policy was announced in summer 2001. Hopefully the effects of the policy direct to comprehensive development in the long run.

The other problem concerns infrastructural deficiencies. For decades the most acute problems have been the poor electricity supply and the poor telecommunications systems (UNIDO 1989, Okuwoga 1990, Ojo 1992, Mursu et al. 2001).

Computer services have increased (see above), but the quality and reliability of these services varies. In the UNIDO (1989) report, as the weakest link was named the area of a reliable maintenance service, partly because of the lack of spare parts (also Akinlade 1989). According to Ojo (1992), IT vendors in Nigeria are more concerned about what they are getting out of a business deal than their client's productive utilisation of the acquired computing system. So he concludes that there is a free-for-all jungle for IT business. He adds that the 'supposed-to-be' IT professional association, the Computer Association of Nigeria (COAN, founded in 1978), which should have appropriately guided the government on this issue, is still struggling to find its right footing. The aims of the association include for example the advancement of information technology penetration and professional practices in Nigeria (Akinde 1998). Akinde continues that one of COAN's visions for Nigerian software industry is

> *"one that will form the bedrock for a supra-national wide co-operation, beginning with the West-African region, and spreading to the development (*developing*) countries. We are determined to facilitate a conducive environment to develop and provide competent, qualified and experienced software houses as well as professionals in Nigeria, with the relevant support and infrastructure for a most rewarding industry".*

The association has organised several national conferences, international conferences, exhibitions, and specialised seminars during the past years.

Even if there is computer education in the country, and it is considered relatively good in terms of hardware and software issues, it is still quite limited when we consider the industrial problem-solving and information systems (UNIDO 1989, Ojo 1992). Ojo asks for curricular flexibility. He points out that there is an apparent unwillingness on the part of the IT educators, who are in leadership positions, to accept an IT perspective that takes them beyond their background in Mathematical and Basic sciences.

The political instability is also affecting the state of IT in Nigeria. The atmosphere of uncertainty and crisis, institutionalised ethnic competition, and the use of state power for wealth accumulation can hinder far reaching development of industry. This also includes economic problems, which can be seen in the falling standard of living, public sector over-size, a weak currency, and the lack of confidence in the financial system. (Sadare 1998)

I am coming back to these issues when presenting the results of our study. It really seems that the main problems have not changed during the years. However, the possibilities to use or acquire IT in Nigeria are quite good for several reasons. Woherem (1992, p. 77) summarises some of them

- huge population
- large human and natural resources
- good educational and skill training establishments
- large markets
- enough financial and human resources to establish viable fibre optics, electronics, and computer hardware industries.

FIGURE 35 An archieving system in a tax office

ISD in Nigeria. Korpela (1994, 1996) studies ISD as a work activity in the case of Medical Records development in the Obafemi Awolowo University (OAUTHC) teaching hospitals complex in Ile-Ife, Nigeria. His study focuses on the central activity of the Medical Records function, and the network of activities, including supporting activities and object activities, according to the activity-theoretical approach. On the activity level, subjects and inter-personal rules were studied. On the organizational level, he studied different stakeholder groups, organizational structures, or organizational culture. He also zooms the level of scope to the societal level. (Korpela 1996)

In his tentative analysis of ISD in Nigeria he suggests that the computer-related activities have two modes. The first he names 'sell and run', concerning hardware sales and businessmen that have little to contribute to the basic needs or national self-reliance. The other trend is based on 'serious entrepreneurs' and 'serious professionals', which pays more attention to systems development. This trend may contribute to an auto-centered or 'delinking' path of development. (Korpela 1996)

In general he proposes that the sustainability of the IS under consideration is determined by the existence or successful implementation of the network of supporting activities. This is in accordance for example with the 'delinking' strategy of national development. (Korpela 1996)

A more important result of Korpela's studies is the elaboration of the activity-theoretical framework which pays attention to the social purpose of the work. In computer related activities, most important after all is the systems development. In my study I have continued this work by using the framework as a research method, as well as the method for analysis. About the framework and Korpela's elaboration, see Chapter 2 in this thesis.

Survey: the profile of software industry in Nigeria. As a part of the INDEHELA-Methods project, a survey was conducted to collect information about software companies in Nigeria (Soriyan et al. 2002). The purpose of the survey was to find out what kind of software industry exists in Nigeria, the type of organization, methods, techniques employed, and the products developed by the software companies and the type of market that is available, also the educational background of the practitioners, and the relevance of the educational institutions in training IS developers for active and effective practice.

The questionnaire was planned to distribute all the identified companies in Nigeria that develop, modify or implement software for customers. Multi-business companies were included if they had software activities. The sample size was difficult to predeterminate, since there is no exact figures of software companies in Nigeria. Related to figures presented earlier in this chapter, we estimated by extrapolation that by the turn of the century, the number of IT companies was at least one thousand, maybe up to 1500. However, probably no more than ten percent – 100-150 – of the companies develop their own software or provide information systems services. The Computer Association of Nigeria (CoAN) and the chartered Computer Professionals of Nigeria (CPN) provided the initial list of software companies to start with. All the 37 states of Nigeria were included in the study. The result was 103 software companies that were included in the analysis. Based on the amount of responses, we can increase our estimate of companies providing software services to 150–200.

The study was conducted during the years 1998 – 2001. It was not possible to send the questionnaires by post (see Methodology chapter in this thesis), so the researchers hired some assistants to deliver the questionnaires and interview the respondents. All in all, with 103 responses, the study provides a good picture of the profile of the software industry in terms of company information, IT personnel, customer companies, products and services, systems development work practices, and projects. (Soriyan et al. 2002)

Most of the companies are still located in Lagos (56%). The Federal Capital City, Abuja, has only 3% of the companies. Most of the companies have only one office in the country (78%). 89% of the companies are Nigerian owned. Most of the companies have been established in the 1990s (62%), 20% in the 1980s. (Soriyan et al. 2002)

Concerning the IT personnel in the companies, most companies (78%) have no more than 15 employees. The companies which have more than 15 employees are usually foreign owned with several offices in the country. The educational background of the employees is at least Bachelor's degree or equivalent. The trend in the software industry seems to be that IT experts obtain a professional qualification in addition to the bachelor degree. The average age of the IT personnel is within 30-39 years. Thus the typical software company in Nigeria is quite small, established in 1990s by Nigerians in Lagos. (Soriyan et al. 2002)

The number of customers per software company varies form 1 to 500. Typically the customer company is a medium-size organization (51-250 employees). Most of the customers are in private sector services (42%), including wholesale and retail trade, transportation, financial services, IT and other business services etc., as indicated in FIGURE 36. Public sector services is the other big customer branch (21%), comprising of health care, education, social work, and administration. Industry (18%) includes e.g. mining, oil, construction, electricity, gas, and water supply, and manufacturing (15%) food, textiles, machinery, equipment etc. (Soriyan et al. 2002)

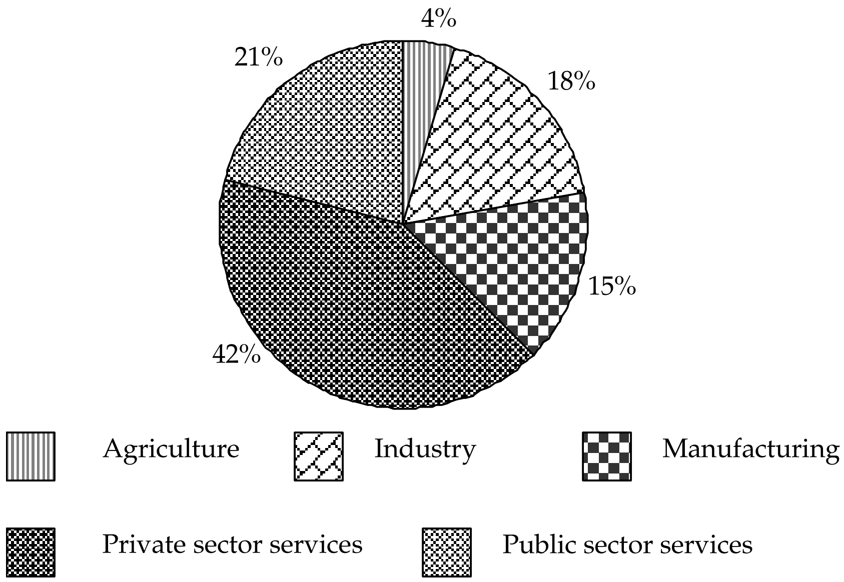


FIGURE 36 Customers' main business activities

The software services the companies provide their customers are divided into imported or self-made applications. Imported packages can be installed as such, or they can be adjusted for customer needs. The same concerns self-made applications. In addition, information systems can be developed from scratch (tailor-made). 45% of the services are based on imported packages, 29% on locally developed packages, and the rest are developed from scratch. One company can provide several such kinds of services. Concerning the platform of the customer, 33% use client-server system with graphic interface, 26% stand alone PCs. Only 13% have web-based systems. (Soriyan et al. 2002)

The most common programming technology is (in 51%) fourth generation language or application generator like Visual Basic, Delphi, JBuilder, Oracle Designer, FoxPro etc. (Soriyan et al. 2002)

Systems development or installation projects takes 6 months on average. 70% of the companies have 2-5 IT professionals involved in a project, 20% have 6-10 professionals, 7% have more than 10 professionals, and 3% have only 1 professional on a project. A typical organization for a project is a formally established implementation team of IT professionals with an internal supervisory board. Usually the project leader comes from the software company. Only 7% have the project leader from the customer company. Usually the project is structured into specific phases, or it is flexible because of the prototyping approach. In systems analysis and design, most companies applied in-house developed methods. About 28% informed to use formal object-oriented methodologies. Also in other phases (like testing and documenting, and project management) the methods are in-house developed. (Soriyan et al. 2002)

In the questionnaire, the possible problem areas in companies' systems development work practices were also asked. The most problematic areas seem to be in risk management, database design, testing, and project planning and management. When compared to problems that the companies point to be in ISD education, their work practices seem to be in pretty good order. According to the respondents, there seems to be very little in the university education that is in good shape. (Soriyan et al. 2002)

In summary, the software companies are not wide spread, which can be partly explained by the limited impact of the Nigerian government in the software industry. In the Federal Capital there are only 3% of the software companies. Also the impact of industry is limited, even though the major export (crude oil) is inclusive. Supposedly the industry deals with foreign software companies, since the majority of their activities are IS dependent. However, it can be hoped that there will be some improvement in the future, especially since there has been recently announced the IT policy by the government. (Soriyan et al. 2002)

Even if the number of customers is large in the software companies, the projects seem to be quite small in terms of time and size. This indicates that on average the installed systems are not the most critical ones. The customers are medium size, although there are some customers with more than 5000 employees (Soriyan et al. 2002). Soriyan et al. observed that the tendency among software companies is to focus on IT training, Internet services and the like, instead of ISD, for economic reasons. Concerning the application packages found in the Nigerian market, most were imported although most were supplied by indigenous software companies.

It is a surprise that the methods applied are mostly in-house developed. This indicates that the methods and techniques in books are not designed for such small companies, or that they are too often only meant for problems amenable to 'clean' solutions. The project organization is usually a formally established team of IT professionals with an internal supervisory board, headed by IT personnel from the software company. This indicates that participation of the customers in the development process is limited. (Soriyan et al. 2002)

The survey indicates that the university education is lagging behind the software industry. According to the companies, their own practice is on average in good shape. So we can assume that the companies have to retrain even the university trained employees, or that the on-the-job training is effective. (Soriyan et al. 2002)

4.4 Summary

This chapter introduces the research context of my study. The study took place in Nigeria, which is located in western Africa. The histories of Africa and Nigeria are briefly examined so that the current situation and problems would be easier to understand and to relate to the industrialised countries. The development theories are also examined so that the goals of IT for development can be realised. It is of no use to make the same mistakes with information technology that has been done with other technologies, resulting in abandoned solutions. However, the gap between developing countries and the industrial countries is still huge, when talking about the use of information and communication systems. In that light the literature of IT in developing countries is examined, and a special notice is given to the appropriateness of IT in developing countries. We argue that at least the most critical applications should be developed or adjusted locally for local needs. So we put special emphasise on the role of local software companies to promote socio-economic or human development with modern IT. The software industry in Nigeria is illustrated at the end of the chapter, the issue that this thesis aims to provide its own contribution.

5 RESEARCH METHODOLOGY AND PROCESS

The purpose of this chapter is to illustrate the research methodology; what kinds of different methods were used and how. The experiences of using different research methods in the research context is also clarified. First, the suitability of different research methods to conduct this study is evaluated, and the research design is explained. Then the risk study is presented. The Delphi method is explained and evaluated. Finally, the ActAd methodology in the case studies is presented, and how the case studies were conducted in practice.

5.1 Suitability of IS research methods to the study

In order to study ISD, researchers need to understand “the ill-structured, fuzzy world of complex organizations” (Avison et al. 1999, p. 95). The choices of practice-related research approaches on IS development include action research (focusing on systems and processes), controlled experiments (field study, laboratory experiment), and practice studies (Mathiassen 1998b). Practice studies are direct, like field and case studies, or indirect, like surveys and interviews. For our purposes practice studies and also action research seem to provide useful methods. I do not consider controlled experiments as an option in our case. Benbasat et al. (1987) summarize that in laboratory experiments the researcher measures dependent variables while manipulating independent variables in a controlled environment. In field experiments the clearly defined variables are manipulated and measured in a natural setting. Also in field studies there are independent and dependent variables, but they are not controlled or manipulated. In case studies, most often there are no *a priori* knowledge of what the variables of interest will be, or how they should be measured (Benbasat et al. 1987).

When the aim is to study the special conditions and requirements for ISD in a previously unstudied context, a *survey* is the easiest way to start with. It provides the researchers with the first contact with companies and practitioners, and provides some first impressions about software industry in the country. It is an attempt to increase predictive understanding of the research question. The survey was a practical choice to conduct the risk study, to which I will come back to later in this chapter. However, a survey alone is insufficient to reach a deeper understanding of the phenomenon of ISD. (Mursu et al. 2000)

For a deeper understanding, the research methodology must be extended to include some *case studies*. Our purpose was to examine ISD as a work activity, using the activity theoretical approach as a research approach and for data analysis (see later in this chapter). Case studies are a proper way to conduct such examinations, where 'how' questions are asked for knowledge building (e.g. Benbasat et al. 1987). Case studies are usually considered as descriptive studies. Besides descriptive, our aim also is to conduct case studies in the interpretative manner. "*Interpretative studies assume that people create and associate their own subjective and intersubjective meanings as they interact with the world around them*" (Orlikowski & Baroudi 1991, p. 5). Hereby, the aim is not to create an objective or factual account of events, but a relativistic, shared understanding of the process of ISD. Instead of a generalization of a population, the aim is to understand the deeper structure of a phenomenon within cultural and contextual situations. Accordingly, ISD as a work activity is examined in its natural setting and from the perspective of the participants. (Mursu et al. 2000)

Surveys and case studies cover a wide variety of approaches to study systems development without the active involvement of the researchers. The weakness with these methods is that they separate research from practice. The researchers observe and interpret the actions and beliefs of the practitioners, and the practitioners do not take an active part in the research process. (Mathiassen 1998b)

A method where the researchers are in close collaboration with practitioners in practice situations is *action research*. The strength of this approach is the strong integration of research and practice, so that research informs practice and practice informs research synergistically (Avison et al. 1999). Action research brings relevance to the research process (Mathiassen 1998b). According to Mathiassen (1998b), the most important weakness is the limited support that action research provides to structuring the research process and findings. Thus the problem is how the rigor of the research can be confirmed. Relatively few sources document the application of action research to systems development. Another weakness, to my mind, is a practical one. However attractive the method would be, it is difficult or impossible to conduct in some cases. It demands the presence of the researchers in the research context for a long time, which is not always possible.

For testing the validity of the theoretical understanding produced through descriptive and interpretative methods, and to test the applicability of the results in real-life ISD practice in Nigeria, action research in one or more software companies is the most relevant way (c.f. Mathiassen 1998b). The idea would be that the practitioners themselves start to influence their work by 'improving' it through reflection, facilitated by the researchers. The research is an iterative process involving researchers and practitioners acting together on a particular cycle of activities, including problem diagnosis, action intervention, and reflective learning (Avison et al. 1999). However, the action research part belongs to the further research activities of the INDEHELA-Methods project. In this thesis, the focus is on survey and case studies, which form a practical part of the research methodology.

5.2 Research design for studying ISD in Nigeria

The methodological discussion above leads to the following overall design of my study (FIGURE 37). The study has three inputs: 1) literature review of related ISD studies; 2) theoretical analysis of the special requirements in Africa and Nigeria; 3) survey about software development risk factors in Nigerian software companies, followed by in-depth case studies focusing on ISD practices. The inputs are used for formulating adapted ISD methods, techniques, and practices specifically for the Nigerian context. This adapted ISD methodology will be distributed to practitioners in Nigeria and Africa through Lecture Notes, which is a documentation of the methodology.

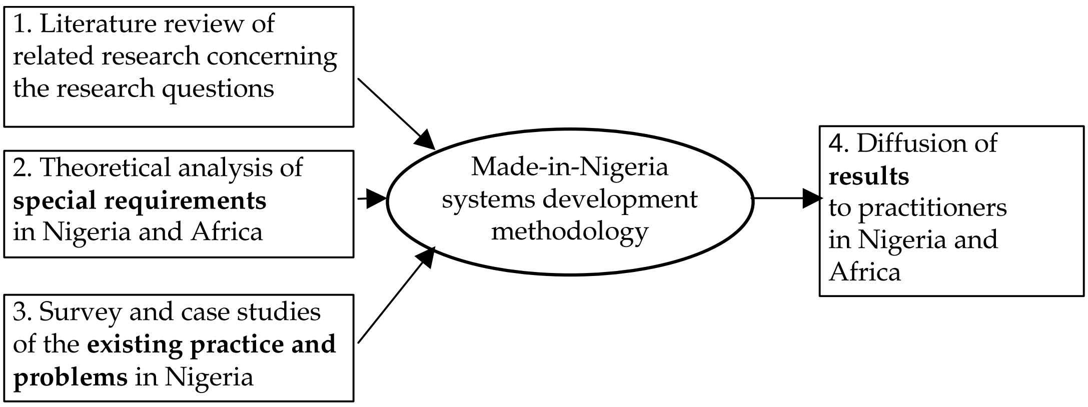


FIGURE 37 The research design.

The overall research design of the INDEHELA-Methods project was introduced in the introduction chapter. FIGURE 37 here illustrates the research actions (inputs) I have conducted during my study. The actual development of Lecture Notes remains after the completion of this thesis.

(1) Literature review of related research has been done in this thesis, as a part of the theoretical foundation. That foundation is based on IS and ISD research, IS research based on the activity theory, and research of developmental informatics in developing countries. The review is related to my research questions concerning information systems development, software development risks, and information systems in use. IS research has been approached by a historical perspective in order to understand the position of this study. In ISD research empirical studies has been examined in terms of the research methodologies and contributions. Several studies of the software project risk factors have been checked for comparison purposes, and studies of risk management in order to get ideas for applied methodology. The use of

information systems has been clarified in the lens of sustainability. Related IS research based on the activity theory has been examined because of the activity theoretical research framework of this study. Finally, since the research is taking place in Nigeria, literature review of information technology, IS development, and ISs appropriateness in developing countries is necessary.

(2) Theoretical analysis identifies areas where the requirements for and constraints of ISD in Nigeria and Africa in general may differ significantly from those of the industrialized countries (Korpela et al. 2000b, Mursu et al. 2000). The purpose is to focus especially on these areas, when developing the adjusted ISD methodology for Nigerian practitioners. The theoretical analysis is based on related literature as well as empirical results. The theoretical analysis is presented in the discussion chapter of this thesis.

(3) The main starting point of the methodological development must be an understanding of the existing practices and problems in ISD in Nigeria, achieved through *empirical research*. We started by an interview study focusing on risk factors in software development projects. Besides the risk factors, we asked some general questions of the companies in order to get some hint of other characteristic features. We continued with the risk study by using the Delphi method, which I will explain next in this chapter. The general questionnaire we enlarged to cover the information systems development in a wider sense, to obtain an overview of the issue. The results of this questionnaire study are presented in the chapter of information technology in developing countries in this thesis. Case studies provide a means to obtain a deeper understanding of the phenomenon.

(4) The three inputs are merged together to an adapted methodology, a portfolio of methods and techniques. This methodology will be diffused to practitioners to be tested and further elaborated. This study provides a contribution for the first version of the methodology, including tools for risk management and sustainability analysis.

5.3 Risk study using the Delphi method

The original idea of my study was to repeat the international Delphi Study conducted in three different socio-economic contexts, viz. Hong Kong, USA, and Finland, by Keil et al. (1998, also Cule et al. 2000, Schmidt et al. 2001). By repeating the Delphi study in Africa, we intend to obtain a richer understanding of the matter by providing a perspective of the developing countries. The aim of the study is to develop a list of risk factors in Nigeria, and to determine which of those risk factors are most important.

In the studies of key issues in IS/IT management in developing countries, the most common research approach used has been the postal survey, either single-round or three- or four-round Delphi studies (Pervan et al. 1998). In the case of studying risk factors, the survey is an obvious choice for the research methodology. One negative point of the survey is that it is likely to obtain little insight into the causes and processes behind the phenomena being studied (Galliers 1991). This shortage will be complemented by case studies. Also there is a possibility of bias from the respondents, at least in posted questionnaires. This will be partly avoided by using personal interviews. Using an interview technique is more expensive, but it is not feasible to use posted questionnaires in Nigeria. By using interviews we ensured the commitment of the respondents and further more, the postal service in Nigeria is not reliable. Besides, according to Marsh (1982), when collecting information about beliefs or attitudes, interviews are the best way. (Mursu et al. 1999)

Keil et al. (1998) chose the Delphi survey as a research method in their risk study in order to get a little more insight into the matter and to elicit the opinions of respondents and to get feedback. In the following the Delphi-method is explained briefly. The Delphi survey process is divided into three phases (Schmidt et al. 2001, Mursu et al. 1999):

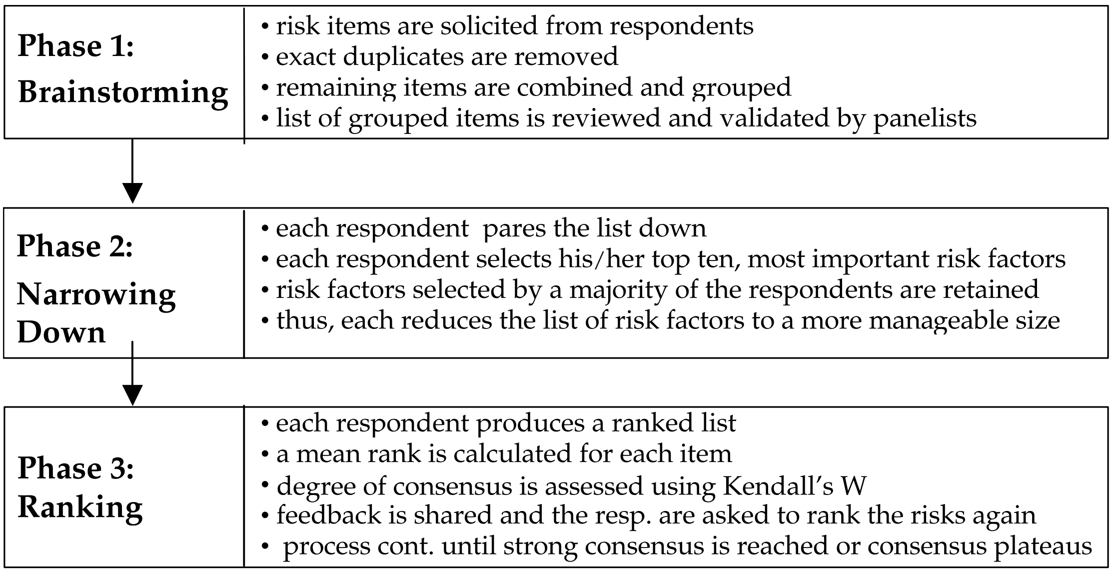


FIGURE 38 Description of Delphi Survey Process Used in this Study

In *phase one*, a brainstorming round is conducted to elicit as many items as possible from the respondents. Each respondent is asked to name at least six factors, and a short description of the factor. The responses are combined and grouped by similar factors. The exact duplicates are removed. The combined list is circulated to all respondents for correction, additions, or concurrence. Schmidt et al. (2001) created this kind of validation for the first phase, which has not been done in other Delphi studies before. In that way they confirmed that each of the three panels (they have panels in Hong Kong, USA, and Finland) are dealing with the same factors, not just 'similar' factors.

In the *second phase*, the list of risk factors is narrowed down to a more manageable and meaningful size to be ranked. Schmidt et al. (2001) separated the three panels, allowing each panel to independently pare down the list of risk factors. Our respondents thus formed the fourth panel. Accordingly, the respondents are asked to choose from the randomized list of items at least ten factors he/she considers the most important, but not to rank these items. The criterion for narrowing the list is that the item selected to be important by over half of the panellists will be retained for the ranking phase.

In the *third phase*, the respondents are asked to rank the risk factors in order of priority, meaning the factors most deserving the project manager's attention. A multiple ranking course might be necessary to reach an acceptable level of consensus. The consensus will be measured by a statistical test called *Kendall's coefficient of concordance W*. The ranking rounds will be sufficient when the respondents reach a strong consensus or consensus does not change from one round to the next. Schmidt et al. (2001) also asked the respondents to rate the risk factors according to their relative importance for the successful completion of a project. This information was used in their analysis of risk management behaviour (Keil et al. 1998).

5.3.1 The Kendall Coefficient of Concordance W

The Kendall Coefficient of Concordance W is an approach where several rankings are conducted for N objects or individuals. When we have *k* sets of rankings, we may determine the association among them by using the Kendall coefficient of concordance W. As a reference for presenting this approach I am using the book by Siegel and Castellan (1988).

When computing W, the data is first arranged into a *k* x N table with each row representing the ranks assigned by a particular judge to the N objects. Then each column (rank) of the table is summed up (R_i) and divided by *k* to find the average rank $\overline{R}_i$. Then these average ranks are summed and divided by *k* to obtain the mean value of the $\overline{R}_i$'s. Each of the average ranks may then be expressed as a deviation from the grand mean. Knowing these values, the value of W can be computed (Siegel & Castellan 1988, p. 264):

$$W= \frac{\sum_{i=1}^{N}(\overline{R}i - \overline{R})^2}{N(N^2-1)/12}$$

k = number of sets of rankings, e.g. the number of judges
N = number of objects (or individuals) being ranked
$\overline{R}_i$ = average of the ranks assigned to the ith object or subject
$\overline{R}$ = the average (or grand mean) of the ranks assigned across all objects or subjects
N(N2 – 1)/12 = maximum possible sum of the squared deviations, i.e. the numerator which would occur if there were perfect agreement among the k rankings, and the average rankings were 1,2,….,N

The value of W varies between 0 and +1, regardless of the number of sets of ranking. As with several nonparametric statistical techniques, the method for testing the significance of the Kendall coefficient of concordance depends on the sample size – in this case, the number of objects being ranked. When N is larger than 7 – as in our case – we must use the table '*Critical values of the chi-square distribution*' for testing the significance. The table is presented in APPENDIX 2.

The quantity $X^2 = k(N - 1)W$ is approximately distributed as Chi square with N – 1 degrees of freedom. That is, the probability associated with the occurrence when H_0 (the assumption that the rankings are independent) is true of any value as large as an observed W may be determined by finding X^2 as presented above, and then determining the probability associated with as large a value of X^2 by referring to the table mentioned above. If the value of X^2 computed equals or exceeds that shown in the table for a particular level of significance and a particular value of $df = N - 1$, then the null hypothesis H_0 that the rankings are unrelated (or independent) may be rejected at that level of significance.

5.3.2 How the Delphi study was conducted in practice

The initial preparations for the Delphi-study started in May 1998. The first version of the questionnaire was formulated during my first visit to Nigeria with our Nigerian colleagues. Besides questions concerning the risks, we asked some general information about the company and its services, and respondents evaluation of the sustainability factors. The idea was to get the first 'hint' of the Nigerian software industry generally at the same time. The questionnaire was tested in May 1998 in the 14th National Conference of CoAN (Computer Association of Nigeria). The corrections were defined during the summer and autumn 1998.

A subset of 11 software companies in Lagos was initially selected for our survey, representing big and small companies, both indigenous and foreign-owned. Some companies produce packaged software, some provide systems development services from scratch or they tailor imported packages for local needs. The companies were selected from the members of the Computer Association of Nigeria by the Nigerian researchers.

As noted earlier, the study was conducted by interviews instead of postal survey. The interviews were aimed at carrying out during the summer and autumn 1998 mainly by some students from the Obafemi Awolowo University (OAU). Nigerian researchers planned to participate the first round in order to start the process with students. There were some delays since the political situation in Nigeria became confused. There was some political turmoil in the country, since the dictator Abacha died and so did the imprisoned president-elect Abiola. There were also ethnic clashes in Ile-Ife. In addition, the everyday duties in teaching of the Nigerian researchers delayed the first round. The situation calmed down and the provisional president Abubakar promised a democratic election during spring 1999.

Phase One. The first interviews were conducted in November 1998 and they were completed in March 1999, when I visited Nigeria for the second time. We collected 39 responses by project managers or comparable persons. We combined our collected risk factors from Nigeria with the list of Keil et al. (1998), before validation. By doing so, we wanted to ensure the comparability of our results with theirs. In addition, there might be some important factors that our respondents did not think of. Also, if these 'additional' factors are not considered as important, they will be dropped out during the next phases. Hence, we considered the combination reasonable.

Not all our respondents were interested in continuing with the Delphi study. Eventually, we agreed with 16 people from eight companies to carry on with the risk study, which was suitable compared to the size of panels in three other countries in previous studies (Keil et al. 1998). All the respondents had an e-mail address, so we hopefully sent the list of combined risk factors to them by e-mail to be validated in the summer 1999. We received some responses to acknowledge the list, but none commenting on it. All in all, only nine of the 16 respondents gave a response by using e-mail at least once during the process, so we found the use of e-mail unreliable and unstable.

Since the interviews turned out to be time-consuming and a laborious method – if the only possible one in that context – we decided to continue straight to phase two. Thus the validation of the collected risk factors remained defective.

Phase two. In autumn 1999 one student from OAU went to Lagos to collect responses for the second phase. He was very thorough and since some of the respondents were not around and some had changed their job, it took two months for him to collect the responses. We could not be in contact with him during that time, so there was a reason to be a little bit worried for his sake. He returned to Ife in November with 24 responses, just before I arrived back Nigeria again. He had taken advantage of visiting the companies again, and increased the number of responses to 24.

In the second phase the panellists each chose the 20 most important risk factors. In the earlier study the researchers chose those factors to be ranked (to continue) which were selected by more than a half of the panellists. In our list only two factors received more than 12 selections. So it was not possible to follow their technique rigorously. I put the limit to those factors that received nine selections or more, resulting for 19 risk factors to be left on the list.

Initially I was hoping to conduct phase three whilst being in Lagos (November 1999), but it was not possible. Since the responses of phase two came so late, we had already agreed to conduct the first case study interviews in Lagos. Also I needed help from the student during the interviews, so nobody was able to go for the risk study and only the analysis of the results was able to be done during that visit.

Phase three. Few students from the OAU collected the responses for the first round of the ranking phase in the beginning of February 2000. They collected answers from eleven participants. Most of the other participants have changed their job or they were out of the office. The agreement of the ranking was very weak, to which I will come back in the results chapter. Hence there was a need for a second round. The second round was conducted by my research colleague and a few students in May 2000. It was delayed a little bit again because of the uneasy situation in the country. For example, there were local ethnical clashes in the Osun state and the road to Ibadan (an old city *en route* to Lagos) was closed for some time. However, the result was six responses, from which one had ties and thus it had to be declined. With five responses we got an even worse consensus when compared to the first round. Since the result did not get any better, we could state that the consensus was not achieved in Nigeria. In addition, since the study had taken too much time already and we had lost most of our respondents, I did not see any reason to continue with the study.

All in all, I can summarize that conducting the three phased Delphi study in Nigeria seemed not to be very practical or a reasonable method. There are several reasons. Firstly, because it took more time than expected, we could not keep our respondents during the process. Even if we used an interview technique, the commitment of the respondents failed. One reason of course was that part of them changed their job. Secondly, the reasons why the study took so much time are distance, resources, and infrastructure. The distance was a reason since part of the researchers live in other countries, and the rest in other towns in Nigeria. Thus conducting interviews in Lagos means travelling. This leads to a resource problem, since it was not possible for researchers to travel all the time when needed, and we had to rely on students. Travelling takes time even inside the mega city of Lagos. This leads to infrastructural problems. It is time consuming to go to Lagos and change places there by using any vehicle, car or public transportation. Sometimes it was only possible to conduct one interview per day. The infrastructure was also the main reason for choosing the interview technique, since the communication network or postal services are too weak to rely on. Thus the method had some limitations in that research context, but we obtained the results and could conclude the risk study.

5.4 Case study – using activity analysis as a framework

The case study is an appropriate method in studies where the problems are ‘sticky’ and practice-based, and where the experience of the actors is important and the context of action is critical. It is applicable to capture the knowledge of practitioners and to develop theories based on it. Also, the case method allows the researcher to ask ‘how’ and ‘why’ questions, which helps to understand the nature and complexity of the studied topic. Finally, the case method is a suitable approach when doing research in an area with few previous studies of the topic (Benbasat et al 1987).

The key characteristics of case studies are the following, according to Benbasat et al. (1987, p. 371).

- Phenomenon is examined in a natural setting.
- Data is collected by multiple means.
- One or few entities (person, group or organization) are examined.
- The complexity of the unit is studied intensively.
- Case studies are more suitable for the exploration, classification, and hypothesis development stages of the knowledge building process: the investigator should have a receptive attitude towards exploration.
- No experimental controls or manipulation is involved.

- The investigator may not specify the set of independent and dependent variables in advance.
- The results derived depend heavily on the integrative powers of the investigator.
- Changes in site selection and data collection methods could take place as the investigator develops new hypotheses.
- Case research is useful in the study of 'why' and 'how' questions because these deal with operational links to be traced over time rather than with frequency or incidence.
- The focus is on contemporary events.

Following the guidelines of conducting case research by Benbasat et al. (1987), we can summarise our study in the following. Firstly, the phenomenon of interest in our study must be studied in a natural setting. Even if there exists a theoretical background for the study, the purpose is more to generate, or elaborate that theory, instead of testing it. There is no reason for controlled or manipulated research actions. As a unit of analysis, we focus on the work activity of information systems development, not just individual, group or organization, but all that it involves in that work activity. All in all, our research question is how information systems are developed in practice, what are the methods, techniques, and practices. By studying the process as a work activity will answer these questions. In order to help identify different elements, one ISD project will be taken as an example in each case. The first priority is not to obtain generalized information, but surely to have representative cases. That is why several cases (multiple-case design) are included. In that way also the elaboration of theory attains more value.

In site selection we tried to be versatile, thus selecting three different types of software companies to be included. The first one is quite a small locally owned company, who developed information systems either from scratch, or they customized their own developed application packages for each customer. The second one is a big foreign owned company, who customized imported packages for local customers. The third one is a small mix-owned company, who develops its own packages for a bigger market, only providing implementation services with slight customisation concerning the platform. All the companies are located in Lagos. This choice is for practical reasons since the interviews were conducted by the researchers and Lagos is the commercial centre of Nigeria, with an international airport. Travelling around the country would not have answered the purpose, since most of the software companies are located in Lagos anyway.

Data collection in case studies usually includes multiple methods (Yin 1984, Benbasat et al. 1987) in order to obtain a rich set of data. The methods include documentation, archival records, interviews, direct observation, and physical artefacts. The Activity Analysis and Development (see Chapter 3) was used as a research framework guiding us in data collection, and we went to the companies with the idea that as many methods as possible will be used for collecting data, depending on the company. The problems or limitations in our research was time and resources. However, there were always two researchers in every interview situation, a Finnish researcher accompanied by a local researcher or assistant. The risk study and the overall survey also included these companies, thus giving some more exact or quantitative data as well.

The analysis of the data is based on the Activity Analysis and Development framework, as was in the data collection. Attention was also given to those issues that had been found important during the theoretical analysis of special requirements, which I will present in the discussion part of this thesis.

In the following, I will present how we applied the ActAD framework in our cases, but before that it might be good to clarify the nature of the cases in this study. Firstly, ActAD is a methodology for work analysis and development. Within this study, we could only use it in order to take 'a snapshot' of the work activity. The real issue would be to continue the process by analysing the contradictions of the activity and the network of activities of the current mode together with the people involved, by examining the history of the activity properly and further, by going on into cyclic development of the activity and the network of activities. Secondly, by using cases we are not trying to analyse and describe one case totally, but to gain knowledge in order to relate and analyse the case in its context, so that it would be able to generalize and use to theory building. The cases will be compared to our survey results.

5.4.1 The research framework - ActAD

In Chapter 3 I have presented the basic structure and the elements of the Activity Analysis and Development framework. As explained, the activity of information systems development can be seen at the border of two organizations. It is a shared activity by a software company or department and user organization, a boundary-crossing temporary activity (Korpela et al. 2002b). The starting point is a problem in the user organization, the need to improve the users' work process. This has been decided to be improved and facilitated by new information technology.

FIGURE 39 presents an imaginary case from Nigeria which is illustrated by using the framework. We used this kind of figure when introducing the research framework to ISD practitioners when we first discussed the cases with them (Soriyan et al. 2000, 2001, Korpela et al. 2002b).

In this imaginary picture, the customer organization, on the right, is a medium size company selling locally designed fashion dresses at the main market in the city of Ibadan, South-West Nigeria. The owner of the company, Chief Mrs. A., is a successful trader, who exports dresses to a Nigerian-born young businessman in London who sells the dresses among the Afro-Caribbean community there. Due to the economic hardship in Nigeria, the export sales activity is an important source for additional income, and Chief Mrs. A. would want to find more customers abroad. However, her export salesperson has great difficulties in trying to communicate with the existing and prospective retailers because of the highly inadequate telephone infrastructure in the country.

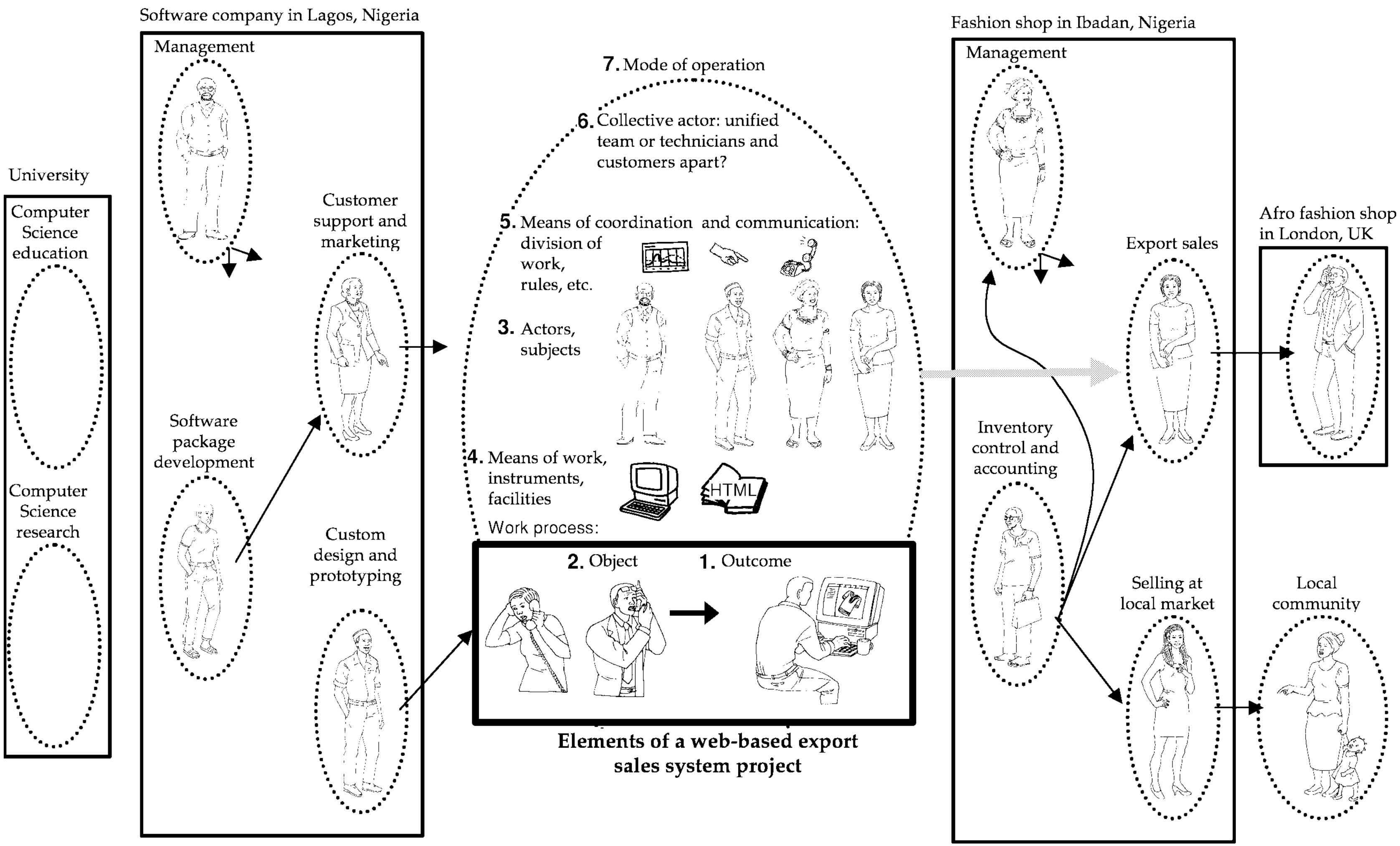


FIGURE 39 Illustration of a sample ISD project, used as research material in Nigeria (Korpela et al. 2002b, draw Reetta Korpela).

In our imaginary case, Chief Mrs. A. had also realized that the manual inventory control and accounting activities in her company were too slow and burdensome for the expanded businesses. She contacted a Dr. O. who owns a small computer company in Lagos and had installed inventory controls and accounting systems to other market traders. The export sales problem was mentioned in passing, and Dr. O. suggested that his technical experts should develop a web-based catalogue that would make Chief Mrs. A.'s wonderful dresses visible and purchasable all over the 'global village'. After all, there are a few Internet Service Providers in Nigeria who hire space for web sites, and they have better telecommunications facilities than ordinary small and medium size enterprises. Chief Mrs. A. accepted the proposal and a project was set up to develop the web-based support system for export sales.

We gave this kind of illustrative examples in the three software companies, with whom we conducted the case studies. With the help of the picture, we could explain to them the issues that we are interested in about their company. We asked them to choose one project in the near past, and we explained that this project can be used as an example of their work. We then started to ask questions like what was the purpose of the project? What kind of people were involved in the project? What kind of skills and development tools did they have at their disposal? From where did they get the skills and tools? What kind of interaction was there between your own staff and the customers, and did you experience any problems in understanding each other? How did you organize the project, what kind of planning and monitoring techniques did you apply? What kind of phases or stages did you experience between the project establishment and its completion, and so on. The basis for the questions is the checklist explained in Chapter 3. However, the questions were applied to be more suitable for a software development project.

5.4.2 How the case studies were conducted in practice

We chose three different kinds of software companies for our case studies; one large foreign owned, one small indigenous and one small mix-owned with its own packaged product. We discussed and agreed on the cases with the company people when we visited in Lagos in March 1999. We promised to provide them with a letter where we explained the purpose of the study, what we expect from the companies and the benefits that the company can get by participating our study.

The results for companies were explained to be twofold: confidential information for the company private use and general information about the branch for benchmarking purposes. The other result was mentioned concerning the education of information technology in universities, since the results are planned for use in developing a better curriculum for information technology studies. For the researchers the benefits were explained to be pure academic, but for the companies the benefits were supposed to be the following:

- self-assessment of the activities with a help of researchers,
- develop practices,

- get knowledge about the branch,
- better services for clients,
- better employees with better education in long view.

Self-reflection and benchmarking were the first benefits we thought the companies would get quite soon. We promised to send them all the articles we would write concerning the issue, as well as the final contributions: thesis and Lecture Notes.

We conducted four case interviews in Lagos in November 1999, at three companies. In the large foreign owned company we had interviews in two different departments. It appeared to be quite difficult to agree on the interviews from Finland by using e-mail or fax. So, during the process we had to rely on the arrangement by our Nigerian colleagues in the first place. However, the interview arrangements changed a few times creating unsure feelings of the success. This was in particular during the last interviews, when I was in Nigeria for two and a half months and we tried to agree on the meetings during my stay.

For the first interviews in November 1999 in Lagos, I received assistance from the same student that had helped with the risk study. Thus he knew the companies already and he proved to be a great help. After the very first interview, we went through the questions we had and restructured them to be more fluent. TABLE 12 provides a structure of the questions, which need to be modified a little to each interview, according to the company in case. The original checklists are presented in Chapter 3.

The interviews were tape recorded, and the noise was a little bit disturbing in most cases, due to the air conditioning or background traffic. Overall, the interviews were very positive and interesting, and they encouraged us to continue the cases and our cooperation with the companies. The average time frame for one interview was three hours. Based on the first interviews there are some lessons to learn:

- If you were a foreigner, take a local colleague with you.
- Because of the language, use a tape recorder in interviews. It is impossible to write down hardly anything. Use a good tape recorder.
- Make a pilot interview beforehand, preferably in your home country in order to test the framework.
- Go through the questions with your partner after the first interview (and also before the first interview)
- Clarify your (all the participants') roles in interview situation
- Become familiar with some interview technique book beforehand.

When I came back to Finland after the first interviews, I started to analyse the cases using the ActAD framework. I took one element at a time and constructed the whole picture of the information system development as a work activity in each case. At the same time the usability of the framework was tested and the framework was further elaborated to be more practical for analysis. The result was a collection of pictures with descriptions of four different cases. I prepared wall graphs of each case in order to use them as help during the next interview situations with the companies. The purpose was to go through the wall graphs in order to find any missing information and correct possible misunderstandings, but the figures were also meant to inspire and direct the following discussions.

TABLE 12 Modified questions for case interviews

1. *The company*: Brief description and history of the company.
2. *The project*: What was the project about? What was the *outcome*? Who was the customer? For what reason did they need your service or product? How did you meet with the customer? How was the *object* of the project defined? Short description of the *process* (phases, schedule etc.)? How was the project organized? What were the roles and responsibilities of each team? How formal was the project structure?
3. *People in the project*: How many people were involved in each part of the project organization? How many men and women? What were their roles and tasks and responsibilities? What was the educational background of the members? What about experiences? Age distribution? What kind of people do you want to employ? What is the procedure and how do you find these people? Does the company provide further education to employees? What kind? What other sources do you use for getting knowledge?
4. *Working in the project*: What kind of professional knowledge or business knowledge was required? What kinds of tools were used? What kinds of methods were used? What kinds of standards were used? Who provides the tools, methods or standards? What was the level of documentation in each phase of the project?
5. *Communication in the project*: About meetings; how formal, how regular, who participates, what kind of issues were discussed and decided? What about other, informal 'meetings'? How decision-making was divided? How were tasks distributed? Did you have any methods or technical means for communication? What was the reporting system? How was feedback distributed? Did everybody know each other in the project organisation? Did everybody know each others' roles? Did you meet outside the project?
6. *Communication outside the project*: Do you think that the user participation was adequate? How did clients participate in the process? How often did you meet and why? How much did you rely on the customer in business issues? How was user training organized? What other stakeholders were involved? What were their roles? How did you communicate with them? How did different stakeholders impact the project? How would you describe top managements' role in the project?
7. *Sustainability*: Is the application in daily use now? Who is using it? Does it improve their services? How critical is the application for the company? How did you evaluate the appropriateness of the technology for the user organization? Did you give any support after the project? Do the customer have an IT department or IT experts? How experienced were the users for the new technology? What do you think are the most important issues for sustainable use of the new system?
8. *Overall*: What was good in the project? What would you do better or differently? What were the major problems? Anything relevant that we have missed? Any questions?

The next round of case interviews was conducted in November 2000. Unfortunately both interviewees at the large foreign owned company had resigned, so we could only visit the two other companies to complete the cases. Thus two cases remained a little bit incomplete. At the other two companies we went through the wall graphs concerning their cases (FIGURE 40). This method worked well to direct the conversation and also to inspire further consideration. We also succeeded to agree on meetings with one of their clients for the following spring. The idea was to meet the client in order to obtain clients' perspective of different research questions. All in all, even if we could not meet the two people again in one of our case companies, being able to complete the two other cases at two software companies and also to agree on the client meetings with them, I felt that my short visit to Nigeria was successful.

FIGURE 40 Interviewing the executive director of Gamma

In spring 2001 I spent two and a half months in Nigeria. The purpose was to meet the clients of our two software case companies, and to discuss the historical dimension of their software development activity. Other goals concerning the research data was to visit a hospital which uses a computer based information system, and to get Wema bank as a case since there has been an IS related research done before (Bada 2000) and we considered that our study would supplement the work by Bada quite nicely. The third additional goal was to organize a work development session in one of our case companies.

Arranging meetings with different people was very difficult and time consuming task. The telephone lines were down most of the time, and we had to travel to Lagos in order to make arrangements. We also had to rely on 'intermediate' people to obtain confirmation of planned meetings. However, we managed to meet a client of the two software companies. The first one was an insurance company and the other was a manufacturing company. We also discussed with an external implementer working for the other software company, being a main contact person between the client and the company. We also visited the software companies again discussing for example the historical dimension of their work. Concerning the extra goals, we managed to visit two hospitals that use information systems. We also visited a consultancy company providing services for example when a client wants to choose the correct software application for their needs. We also faced some failures. We could not get the Wema bank as a case. We also did not manage to start the work development process at one of the software companies, because of the time limits, even if they were willing to do that in principle. However, the other company started to reflect their activities by themselves, and they reported to us on their progress.

All in all, we managed to have two in-depth case studies from local software companies, two incomplete case studies from a foreign owned software company, and two interviews with client representatives. As additional data, giving a nice addition and extra ‘colour’ to the cases, we had one interview with a consultancy company, one interview with an external implementer working for the other local software company, and two hospital visits, being acquainted with their information system. However, I will only report on the implementer interview explicitly in the results chapter, the others are treated as general background knowledge.

Concerning the data collection, the cases remained somewhat limited when compared to what was planned, including mainly interview data. In some cases we also obtained data from some technical documents or more general brochures. However, the interviews were very open ended and dynamic providing much information.

6 NIGERIAN CASES – THE RESULTS

In this chapter the results of the study are presented; the risk study and the in-depth case studies. In order to better understand how the risk factors are related to software development in Nigeria, we must know how the systems are developed in practice. Thus a few in-depth case studies were conducted to find out the practices and techniques applied in ISD work in Nigeria, since this kind of research has not been done earlier in Africa.

The risk study was conducted by using the three-phase Delphi method. The results of each phase are presented and the ranking phase is discussed briefly by comparing it to earlier risk studies conducted in industrialised countries. The case studies are presented and analysed after that. Finally, I sum up the results of the two research methods with a survey that has been made at Nigerian software companies.

6.1 Risk study

The risk study repeats an international risk study conducted earlier in the US, Finland, and Hong Kong (Keil et al. 1998). Thus the study also follows the methodological design by using the three-phase Delphi method as a research method. The results of each phase are introduced in the following. The result of the risk study have been reported earlier in articles of Mursu et al. (1999) and Mursu et al. (submitted).

6.1.1 Phase one

We received 39 responses from eleven software companies in Lagos during phase one of the Delphi-study. Eight of the interviewees were women. The companies mostly represented Nigerian owned private companies, two companies were foreign owned. Almost all the companies, except one, produced systems for outside clients, like banks or the oil industry, and they mostly produced business and management systems. One company produced its own packages. The following table (TABLE 13) shows major demographic information about the participating companies, with comparison data concerning the companies that participated in the study in the US, Hong Kong (HKG), and Finland (FIN) (Keil et al. 1998). Since all the companies were software companies (no in-house departments), there is no specification about the total employees and IS employees in Nigerian companies. The last column indicates the composite information of the US, HKG, and FIN.

TABLE 13 Demographic information of the study subjects of Nigeria, the US, Finland (FIN) and Hong Kong (HKG) (adapted from Schmidt el al. 2001, p. 12)

Characteristic	**Nigeria**			**US**	**FIN**	**HKG**	**Composite of US, FIN and HKG**
	Avg	Max	Min	Avg	Avg	Avg	(Avg)
Total Employees in Panelist's Company	112	500	12	10574	3693	3039	6616
IS Employees in Panelist's Company				292	156	101	203
Panelist's Work Experience (Years)	9	21	2	15.3	20.1	13.2	16.
Panelist's Educational Level	PS	PD	BD	BD	PS	BD	BD
Number of Projects Panelist has Managed	7	20	1	35.8	13.5	10.3	22
Smallest Project Managed by Panelist (Person Months)	3.25	12	0.25	14.4	28.1	16.6	19
Largest Project Managed by Panelist (Person Months)	18.25	65	3	887.6	484.1	335.6	633

'Education Level' is the highest level attained by the panellist, the scale:
HS = High School
HD = Higher Diploma (usually one or two years past high school)
AD = Associate Degree (two years of university-level education)
BD = Bachelor's Degree
PS = Some university study beyond the Bachelor's Degree
PD = Postgraduate Degree (Master's or Doctorate)

It can be noted that Nigerian companies are smaller (they were all software companies), the work experience of the respondents is less on average, and the projects are smaller in Nigeria when compared to the averages of other countries. This affects somewhat the comparability of the results, but it must be remembered that software industry in Nigeria is young (Soriyan et al. 2002), and there are rare foreign investors, which affects the size of the companies.

The interviews collected 131 separate risk factors before modification. We combined and organised the factors according to categorisation of Keil et al. (1998, reported more detailed in Schmidt et al. 2001) since the list is too long to be managed in a non-hierarchical form. Schmidt et al. (2001) used classification of 14 categories. We included one additional group of risks – that of a socio-economic context – because we obtained risk factors differing considerably from the other factors resulted in earlier studies (TABLE 14). Schmidt et al. define their first group – corporate environment – to also contain changes in the political environment, but none of their risk factors focused on the socio-economic context in the country. In contrast, most of the risk factors in our group focused on the general socio-economic situation rather than the corporate environment. Therefore we decided to form a new group called a *socio-economic context.*

TABLE 14 Risk categorization, modified from categorization of Schmidt et al. (2001, p. 35)

Risk Category	Source of Risk, Nature of Risk
0. Socio-Economic context	**Context:** Condition of the political environment.
1. Corporate Environment	**Environment:** Changes in the business or political environment or poor alignment of the system with the organizational culture.
2. Sponsorship/Ownership	**Mandate**: Lack of mandate for the PM to execute the project plan. Lack of trust or poor relationships with the owners of the system.
3. Relationship Management	**User Relationships**: Lack of trust and inadequate user involvement. Unclear roles and expectations among users or other stakeholders.
4. Project Management	**Management**: Poor or inefficient management strategy and execution.
5. Scope	**System Scope**: Unclear, changing or partial understanding of the system scope and mission.
6. Requirements	**Requirements**: Inadequate or poor management of system requirements; poor validation of system requirements.
7. Funding	**Resource management**: Too little or badly estimated resources for SD.
8. Scheduling	**Resource control**: Poor management of resource consumption and needs. Poor timing.
9. Development Process	**Process**: Inappropriate or lacking process approach.
10. Personnel	**Skills**: Inadequate personnel skills in development and process management.
11. Staffing	**Staffing:** Changes in personnel or staffing levels, unavailability of key personnel resources.
12. Technology	**Technology:** Inadequate understanding of the chosen technology.
13. External Dependencies	**Development environment**: Poor management or control over dependencies with external agents.
14. Planning	**Planning**: No interest or inadequate skills to plan the project.

The list of risk factors identified in Phase 1 is shown in TABLE 15. We ended up with 51 different risk factors. In TABLE 15, after each factor there is a number indicating the occurrence of the risk factor in the total amount of 131 items, thus showing the frequency of the factor. The shaded items represent factors that are not represented in risk lists of earlier studies. The order and numbering of the factors are based on the risk list of Schmidt et al. (2001, p. 15-18).

The examination of the frequencies in risk factor groups indicates that most often mentioned risk items are in the group of a socio-economic context, relationship and project management, requirements, funding, personnel, and staffing. The groups of the middle size consist of sponsorship / ownership and technology. The groups with the smallest number of items are in corporate environment, scope, scheduling, development process, external dependencies, and planning.

TABLE 15 Collected risk factors in phase one (adapted from Schmidt et al. 2001, p. 15-18).

0.	**Socio-economic context**	**Fre**
0.1	*Political climate in the country, including economic situation*: the poor economical state of the country does not allow for IT improvement and investment	5
0.2	*IT awareness in the country*: lack of proper IT exposure and policy	1
0.3	*Erratic and unreliable communication network*: poor communication (network, telecommunication etc.) can hinder some projects	4
0.4	*Energy supply*: When regular supply of power to computer systems cannot be guaranteed	5
0.5	*Tertiary institutions*: tertiary institutions in country today lack a lot of facilities required to prepare student for solid IT future	1
0.6	*Poor copyright / intellectual property right protection*	1
1.	**Corporate Environment**	
1.1	*A climate of change in the business and organizational environment that creates instability in the project*	1
1.2	*Unstable Corporate Environment:* Competitive pressures radically alter user requirements, sometimes making the entire project obsolete.	1
2.	**Sponsorship/Ownership**	
2.1.	*Lack of Top Management Commitment to the Project*. This includes oversight by executives and visibility of their commitment, committing required resources, changing policies as needed	2
2.2	*Lack of client responsibility, ownership, and buy-in of the project and its delivered system(s).*	1
2.3	*Failure to gain user commitment*: Laying blame for "lack of client responsibility" on the project leader rather than on the users.	2
2.4	*Customer's staff turnover*	1
3.	**Relationship Management**	
3.1	*Failure to Manage End User Expectations*: Expectations determine the actual success or failure of a project. Expectations mismatched with deliverable - too high or too low - cause problems. Expectations must be correctly identified and constantly reinforced in order to avoid failure.	2
3.2	*Lack of Adequate User Involvement*: Functional users must actively participate in the project team and commit to their deliverables and responsibilities. User time must be dedicated to the goals of the project.	5
3.3	*Lack of Cooperation from Users*: Users refuse to provide requirements and/or refuse to do acceptance testing.	2
3.5	*Growing Sophistication of Users Leads to Higher Expectations*: Users are more knowledgeable, have seen sophisticated applications, apply previous observations to existing project.	1
3.7	*Lack of appropriate experience of the user representatives*: Users assigned who lack necessary knowledge of the application or the organization	1
3.8	*Customer's ability to react to change*: Certain individuals do not want to conform the change because of general phobia of computers or they are worried their jobs are threatened	4
3.9	*Inadequate user training*: adequate training of customer is required to maintain systems.	1
3.10	*Negligence of agreements*: clients make illegal duplicates of packages	1
4.	**Project Management**	
4.1	*Not Managing Change Properly:* Each project needs a process to manage change so that scope and budget are controlled. Scope creep is a function of ineffective change management and of not clearly identifying what equals success.	2
4.2	*Lack of Effective Project Management Skills*: Project teams are formed and the project manager does not have the power or skills to succeed. Project administration must be properly addressed.	4
4.3	*Lack of Effective Project Management Methodology*: The team employs no change control, no project planning or other necessary skills or processes.	2

(Continues)

TABLE 15 (continues)

4.4	*Improper Definition of Roles and Responsibilities*: Members of the project team and the organization are unclear as to their roles and responsibilities. This includes outsourcers and consultants.	1
4.6	*Poor Risk Management*: Countering the wrong risks.	1
5.	**Scope**	
5.1	*Unclear/Misunderstood Scope/Objectives*. It is impossible to pin down the real scope or objectives due to differences or fuzziness in the user community.	1
5.3	Scope Creep: Not thoroughly defining the scope of the new system and the requirements before starting, consequently not understanding the true work effort, skill sets and technology required to complete the project.	3
6.	**Requirements**	
6.1	*Lack of Frozen Requirements*. Because the needs of the users change, the requirements change. Consequently the system will never be moved into production because none of the requirements are ever completed. Alternatively, freezing a subset of the functionality and delivering allows for the completion of the system and update releases as required.	5
6.2	*Misunderstanding the Requirements*. Not thoroughly defining the requirements of the new system before starting, consequently not understanding the true work effort, skill sets and technology required to complete the project.	3
6.3.	*New and/or Unfamiliar Subject Matter for Both Users and Developers:* Lack of domain knowledge leads to poor requirements definition	2
6.4	*Inadequate documentation of user requirements*	4
7.	**Funding**	
7.1	*Under Funding of Development*: Setting the budget for a development effort before the scope and requirements are defined or without regard to them (i.e., picking a number out of the air).	11
7.3	*Bad Estimation*: Lack of effective tools or structured techniques to properly estimate scope of work. Unrealistic cost estimates cause illogical or sub-optimal planning, strategy, and decisions.	1
7.5	*No investments to IT*: Investors are cautious to put money on software development.	1
7.6	*Huge capital requirements*: There is no efficient leasing program in the country (hardware is expensive)	1
7.7	*Poverty of software companies*: employing staff is very expensive, and the wages serve as catalyst for them to be committed	1
8.	**Scheduling**	
8.1	*Artificial Deadlines*. Presence of unrealistic deadlines or functionality expectations in given time period. - 'crash projects' in which test time or training time is reduced – using something other than work effort required to determine when the new system should move into production.	3
8.3	*Lack of experience of the user management*: timing demands are unrealistic	1
9.	**Development Process**	
9.1	*Lack of Effective Development Process/Methodology:* Leading to quality problems - Documentation, Software and Testing—poor estimating -- insufficient time for up-front work, e.g., design—little flexibility for change—insufficient testing.	2
9.3	*Lack of quality standards*	1
9.4	*Lack of computing literature*: no literature for development process or techniques	1
10.	**Personnel**	
10.1	*Lack of Required Knowledge/Skills in the Project Personnel*: e.g., technology, business knowledge and experience.	7
10.2	*Lack of "People Skills" in Project Leadership*: PM tries to "manage" schedules, technology, requirements, etc., ignoring that management is dealing with people on the team.	1
10.3	*Poor Team Relationships*: Strains existing in the team due to such things as burnout or conflicting egos and attitudes.	5
11	**Staffing**	
11.1	*Insufficient/Inappropriate Staffing*: Not enough people or people with wrong skills/insufficient skills assigned to project, regardless of availability.	1
11.2	*Staffing Volatility*: At some point in the project, losing the key project manager, analysts or technicians (especially in new technology).	5

(continues)

TABLE 15 (continues)

11.4	*Lack of Available Skilled Personnel*: People with the right skills are not available when you need them.	4
12.	**Technology**	
12.1	*Introduction of New Technology:* Using new, or 'bleeding edge', technology that has not been used successfully at other companies, or major technological shift occurs during the project.	2
12.3	*Inappropriate technology*: Trying to achieve a particular task/project without the appropriate tools	7
13.	**External Dependencies**	
13.4	Importation of foreign packages: craze for foreign packages with cheep price do not allow the growth of indienous developers	2
14.	**Planning**	
14.1	*No Planning or Inadequate Planning*: Attitude that planning is unimportant or impractical.	**4**
		131

Note: Items are grouped by category. Shaded items represent risk factors not observed in earlier lists (Schmidt et al. 2001).

To render the results more comparable with earlier studies, we combined our list with the list of Schmidt et al. (2001). They compared their list to a merger of older risk item lists by Barki et al. (1993) and Boehm (1989) and concluded that it covered all risks observed earlier and thus it was more encompassing. We used this combined list (see Appendix 3) in the next phase of the Delphi study to see if some important risks were not considered during the first Phase of this study. Overall, our merger of the lists resulted in 20 risk factors that have not been mentioned in previous studies (shaded items in TABLE 15). Thus, the new risk factors obtained in this study increase the coverage of relevant risks. Accordingly we found 31 similar risk factors from earlier studies. 21 items in Schmidt et al. list were not identified in the Nigerian study. A systematic comparison to Schmidt et al. list is shown in Appendix 4.

6.1.2 Phase two

The result of phase two of narrowing down the risk list to a more manageable size, is identified in TABLE 16. In this second phase the participants were asked to choose 20 most important risk factors.

In the earlier study the researchers chose those factors to be ranked in the next phase which were selected by more than half of the panellists. In our list only two factors got more than 12 selections (24 participants). So it was not possible to follow their technique exactly (Keil et al. 1998). We put the limit to those factors that got nine selections or more. We ended up with 19 risks. What is noteworthy is that three of the not recognized factors were at the end selected amongst the most important 19 risk factors when we used the combined list (Appendix 2) in phase 2. These were 'Choosing the Wrong Development Strategy', 'Changing Scope/Objectives', and 'Trying New Development Method/Technology During Important Project'.

TABLE 16 Selected risk factors for ranking phase.

Risk factors in the order of number of selections	Selections
Erratic and unreliable communication network	14
Artificial Deadlines	13
Energy supply	11
Inadequate user training	11
Unclear/Misunderstood Scope/Objectives	11
Lack of Required Knowledge/Skills in the Project Personnel	11
Lack of Available Skilled Personnel	11
Failure to gain user commitment	10
Choosing the Wrong Development Strategy	10
Misunderstanding the Requirements	10
Under Funding of Development	10
Huge capital requirements	10
Lack of Effective Development Process/Methodology	10
IT awareness in the country	9
Lack of Top Management Commitment to the Project	9
Changing Scope/Objectives	9
Trying New Development Method/Technology During Important Project	9
Lack of "People Skills" in Project Leadership	9
Importation of foreign packages	9

6.1.3 Phase three

For a final phase, the list of over 70 factors was reduced into 19 risk factors to be ranked by the participants. To analyse the rankings we used Kendall coefficient of concordance W in order to find out the significance of the consensus.

TABLE 17 indicates the result of the first round. In the table there are the average ranks of the items and the calculated values for Kendall's W, which is calculated in the last row. The formula for the calculated values for W in each row is based on Schmidt's macro (Schmidt, personal communication 29th of September 1999, see also Schmidt 1997):

*Av.rank – SUM(Av.ranks)/15)*2*

The Kendall's coefficient of concordance W has a formula by Schmidt:

*SUM(Values for W)/(15*2-1)/12)*

The number of N (items to be ranked) in the risk study was 19, and the number of *k* (responses) in the first round was eleven. The *k* x N table brought the following result:

Then the value of X^2 was computed, which is approximately distributed as Chi Square.

$X^2 = k(N\text{-}1)W = 11(19\text{-}1)0.142 = 28.116$

The significance with df = N – 1 was tested using the table in Appendix 2. The significance is p = 0.10, which means that the null hypothesis will remain and the rankings are unrelated. The consensus was very weak.

TABLE 17 The first round of ranking

Risk factor	Round 1 Av.rank	Value for W
Misunderstanding the Requirements	5.090909	23.63169
Under Funding of Development	7.545455	5.792198
Artificial Deadlines	7.727273	4.950093
Lack of Effective Development Process/Methodology	7.818182	4.553833
Unclear/Misunderstood Scope/Objectives	7.909091	4.174103
Erratic and unreliable communication network	8.454545	2.242829
Lack of Required Knowledge/Skills in the Project Personnel	9	0.906596
Lack of Available Skilled Personnel	9.272727	0.461619
Changing Scope/Objectives	10.18182	0.052746
Choosing the Wrong Development Strategy	10.36364	0.169318
Lack of Top Management Commitment to the Project	10.54545	0.352007
Inadequate user training	10.63636	0.468144
Failure to gain user commitment	11.36364	1.992285
Energy supply	11.72727	3.15105
Huge capital requirements	11.90909	3.829606
Importation of foreign packages	11.90909	3.829606
IT awareness in the country	12.09091	4.574277
Trying New Development Method/Technology During Important Project	12.63636	7.204986
Lack of "People Skills" in Project Leadership	12.90909	8.743481
Kendall's W		0.142246

Thus, we needed another round. In that round the number of k (respondents) decreased to six. One of the respondents used ties, so the response had to be disqualified. Now we had the number of N = 19 and k = 5. The result is presented in TABLE 18.

Then the value of X^2 was computed, which is approximately distributed as Chi square.

$$X^2 = k(N-1)W = 5(19-1)0.255579 = 23.00211$$

The significance with df = N – 1 was tested using the table in Appendix 2. The significance p = 0.20 was even weaker than in the first round. Thus the null hypothesis still remains and the rankings are unrelated. The consensus became weaker.

In our study we did not receive a significant value of W. We can order the risk factors by their average ranks, but we cannot interpret that this is the order of importance in Nigeria. The most important result obtained by this ranking is the reduced list, which covers those risks that were considered to be most important by the Nigerian project managers (19 risk factors). Overall, we can say that this list reflects the current and the most important barriers to successful software development in Nigeria.

TABLE 18 The second round of ranking

Risk Factor	Round 2 Av.rank	Value for W
Misunderstanding the Requirements	4.4	31.36
Lack of Effective Development Process/Methodology	4.8	27.04
Lack of Required Knowledge/Skills in the Project Personnel	5.8	17.64
Lack of Available Skilled Personnel	7.6	5.76
Under Funding of Development	8.6	1.96
Importation of foreign packages	9.4	0.36
Lack of "People Skills" in Project Leadership	9.6	0.16
Unclear/Misunderstood Scope/Objectives	10	0
Changing Scope/Objectives	10	0
Energy supply	10.2	0.04
Artificial Deadlines	10.2	0.04
Inadequate user training	10.6	0.36
Choosing the Wrong Development Strategy	11.4	1.96
Lack of Top Management Commitment to the Project	11.4	1.96
Failure to gain user commitment	11.8	3.24
IT awareness in the country	12.4	5.76
Huge capital requirements	13	9
Erratic and unreliable communication network	14	16
Trying New Development Method/Technology During Important Project	14.8	23.04
Kendall's W		0.255579

It should be emphasized that a high or significant value of W does *not* mean that the orderings observed are *correct*. In addition, a high degree of agreement about an ordering does not necessarily mean that the order agreed upon is the '*objective*' one. Kendall suggests that the best estimation of the 'true' ranking of the N objects is provided, when W is significant, by the order of the various sums of ranks R_i or the average rankings $\overline{R_i}$. (Siegel & Castellan 1988)

We also indicate the relative importance of each risk in addition to the final ranking (TABLE 19). We asked on top of ranking the criticality to rank the risk factors in terms of their importance to control for the validity of the first ranking.

6.1.4 Discussion of results

By repeating the earlier study, which was conducted in the US, Finland, and Hong Kong, in a different context in Nigeria, we expected some notable differences in observed software risks and new factors due to radically different socio-economic and infrastructural environment. When we compared our risk list of 51 risk items with the list produced by Schmidt et al. (2001), there are some expected differences. Initially we expected that the differences would be more pronounced in areas that are affected by radically different economical and technological environment, and by some obvious cultural differences. Moreover, we expected to see some unique items that have not been detected in earlier studies within the industrialized countries, because our initial interviews revealed that the risk focus is on the economical, social, and the infrastructural issues.

TABLE 19 The relative importance of risk factors

#	Risk Factor	Importance: Round 2
1.	Misunderstanding the Requirements	8.1
2.	Lack of Effective Development Process/Methodology	7
3.	Lack of Required Knowledge/Skills in the Project Personnel	6.6
4.	Lack of Skilled Personnel	7.1
5.	Under Funding of Development	7.2
6.	Importation of foreign packages	6.5
7.	Lack of "People Skills" in Project Leadership	6.7
8.	Unclear/Misunderstood Scope/Objectives	7.6
9.	Changing Scope/Objectives	7.2
10.	Energy supply	6.5
11	Artificial Deadlines	7.4
12.	Inadequate user training	6.6
13	Choosing the Wrong Development Strategy	5.8
14.	Lack of Top Management Commitment to the Project	6.7
15.	Failure to gain user commitment	6.5
16.	IT awareness in the country	5.4
17.	Huge capital requirements	5.9
18	Erratic and unreliable communication network	6.8
19.	Trying New Development Method / Technology During Important Project	5.6

When compared, our risk list and the list by Schmidt et al. (2001), we found some similarities as well as differences. Firstly, there are 21 risk factors in Schmidt et al. list that are not identified in our list. Many of the factors *not identified* in our study are located on the *business strategy level*. They mostly deal with business processes, stakeholders, possible other vendors and consultants (see a systematic comparison in Appendix 3). These can be seen to reflect a lower level of alignment between software development and business development in Nigeria. Interestingly, two of these factors, 'Conflict between user departments' and 'Changing scope/Objectives' (which was later noticed in Nigeria as well), were selected among the eleven most important risk factors in the US, Finland, and Hong Kong.

Schmidt et al. (2001) compared their results to a merger of older risk item lists by Boehm (1989) and Barki et al. (1993). They noticed in their study a new risk topic dealing with the *diversity and multiplicity of stakeholders and user communities* (e.g. 'Failure to identify all stakeholders'). This factor is missing in our study, like the *evolution of IT infrastructure* ('Stability of technical architectures'). Instead the lack of basic IT infrastructure was emphasized. This points out that software and business solutions in Nigeria are not that mission critical to the same extent and encompass a smaller set of constituencies and utilize a small range of services from the existing IT infrastructure. At the same time, *critical aspects in the management environment*, which Schmidt et al. (2001) found to be lacking in previous studies, was also identified in Nigeria. These include: 'Poor risk management', 'Choosing the wrong development strategy', and 'No planning or inadequate planning'. These themes reflect the dynamic nature of IS development in Nigeria, and the challenges it creates for project managers (Schmidt et al. 2001). One reason for its occurrence in Nigeria can be the short history of software development. Although the software companies in Nigeria are ambitious, the history of indigenous systems development in Nigeria is short. As one of the managers said, "*there is no tradition in computer business*".

We also observed new factors that the previous study did not notice, 20 items altogether (see the shaded factors in TABLE 15). These factors, in fact, are not *new*, since many of the issues are familiar to anybody who knows the development literature (see e.g. Waema 1996, Roche & Blaine 1996). Many of these risk items focus on the socio-economic context of the country including its poor communication network and electricity supply. Accordingly, companies need to buy alternative sources of electricity that also break down easily. As one of the interviewees described "*even when the generator works, fuel is sometimes impossible to procure except on the black market. When NEPA* (National Electric Power Authority) *does provide power, it is often so bad that computers re-boot on their own*". The same holds with the availability of reliable telephone lines. One manager told us that when they tried to get six lines to their office, they spent two years and $1.500 per line to get them. Yet they are happy if they have three lines working at the same time despite the fact that they have a person half the time looking after the telephone lines.

The lack of national IT policy and IT investments establish general constraints for IS development. IT investments need a lot of additional money for maintenance of the infrastructure they require, and many resources have to be pooled from companies. Moreover, these resources are often scarce. In addition, the unstable political situation can affect the business directly. As one manager bluntly phrased it: "*The government is the biggest buyer in the community, and when the government stops operating it has a ripple off effect to all other business as well*." In a speech by the government representative in a software exhibition in Lagos 2001, the national IT policy was promised to be worked out in near future.

Another notable group of new risks focuses on user relationships with inexperienced users. This is due to low levels of IT literacy in Nigeria. These risks are close but not similar to Boehm's (1989) 'unwilling users' and 'user resistance'. User relationships concern both the management level and the user level. Inexperienced users need plenty of post-implementation support, and this becomes expensive for companies. Yet, the most common user relationship – risk identified in Nigeria is the same as in the earlier study- 'Failure to gain user commitment' or 'Lack of adequate user involvement' (Schmidt et al. 2001). Accordingly, there is a need for managing user relationships and user expectations. Today users must take more responsibility over the system development and be involved in a host of development activities. We observed 'Customer's staff turnover' as a risk factor, which has not been mentioned in other studies, although Schmidt et al. observed 'Lack of appropriate experience of the user representatives'. Barki et al. (1993) mentioned 'User turnover' in their study.

The rest of the new risks concern the development process, its management, and knowledge bases like 'Inadequate documentation of user requirements', 'Lack of quality standards', 'Lack of the computing literature', or 'Inappropriate technology'. Most of these are critical aspects on the project management level. 'Tertiary institution' refers to poor university education, which suffers from the lack of facilities and provides no access to the latest knowledge.

Many of the new issues can be seen to be outcomes of the socio-economic and infrastructural context, and the relatively low level of maturity of the computerized activities in Nigerian organizations. Like in earlier studies, there were, however, only a few items on the list that handle the *technical* aspects of software development like the choice of programming languages, operating system compatibility, or user interfaces (Schmidt et al. 2001). These problems are, however, often reflected in the inadequate technical skills of the personnel, which is many times a result of poor or the lack of education, as well for example in the risk factor 'Inappropriate technology'.

The most obvious 'new' set of risk factors identified by Schmidt et al. (2001) and also in our study is inadequate *project management methodologies and project management skills*. This echoes the growing awareness of the importance of putting the correct project management practices in place in Nigeria, and the risks associated with failing to do so.

All in all, our study highlights some new dimensions in software risk management, that have not been recognized in the previous studies conducted in highly industrialized countries. The combined list of risk factors from the studies in US, Finland, Hong Kong, and Nigeria (see Appendix 3) helps to organize knowledge of the types of risks the project managers will face all over the world. Together with the categorization provided (see Schmidt et al. 2001) it can offer valuable support for project managers in assessing project risks in different socio-economic situations. Our addition from Nigerian risk management experience brings socio-economic environment more visible as one risk source.

In the second phase, where the combined list of risk factors was narrowed down, three of the not recognized factors were at the end selected amongst the most important 19 risk factors. These were 'Choosing the Wrong Development Strategy', 'Changing Scope/Objectives', and 'Trying New Development Method/Technology During Important Project'. This stresses the importance of using as encompassing list as much as possible as a basis for developing the list to be ranked.

Risk ranking. In the final phase, we conducted two rounds of rankings, during which we did not receive a significant value of W. Due to the type of metric we use we can order the risk factors by their average ranks, but we cannot interpret that this reflects the final order of importance. However, the most important result obtained by this ranking is the reduced list, which covers those risks that were considered to be the most important by the Nigerian project managers.

Those factors that stand out in ranking averages deal with 'Misunderstanding the requirements', (Avg rank = 4.4), 'Lack of effective development methodology' (4.8), 'Lack of required skills and knowledge' (5.8), and 'Lack of skilled personnel' (7.6). Thus the requirements, control over the development process, and the lack of skills and personnel are the most critical risks that have to be managed by the Nigerian project managers. This list is somewhat different from the earlier study in that the issue of top management support, which is very pronounced in earlier studies is not recognized at all as the top risk item.

Even with a low consensus, we can compare the rankings of our study and the study of Schmidt et al. (2001). They argue that factors are situation dependent and that sampling from a single culture tends to leave us blind to some important risk factors. Thus it is difficult to generalize one simple 'top-ten' list of risk factors across different cultural settings. Accordingly, when we compared the ranked factors with earlier ranked factors lists some factors were very different from those included in their list.

TABLE 20 summarizes all rankings obtained in the four countries. Altogether, Nigeria had 19 items, while Hong Kong had 15, Finland 23 items, and USA 17 items. All other countries except Nigeria reached some level of consensus in ranking, though the obtained rankings do not match particularly well with each other. Among the US, Finland, and Hong Kong there were eleven items that all these countries had in common (Keil et al. 1998). If we intersect this list with the Nigerian list the number of common items is reduced to five items.

We will first consider the five items that all four countries included into their rankings. These are 'Misunderstanding the Requirements', 'Lack of Required Knowledge/Skills in the Project Personnel', 'Changing Scope/Objectives', 'Lack of Top Management Commitment to the Project', and 'Failure to gain user commitment'. 'Lack of Top Management Commitment' was chosen as a top choice almost by common consent by the IT experts in the US, Finland, and Hong Kong. It was recognized also in Nigeria within the top 19 items (14^{th} in ranking). As Schmidt et al. (2001) observed, top management must play a strong and active role in projects from the initiation through implementation despite variance in the socio-economic context. Two other factors that received a high rating in all three other countries were: 'Failure to gain user commitment' and 'Misunderstanding the requirements'. 'Misunderstanding the requirements' was ranked as the top item in Nigeria. Notably, in all countries it was included in the top 10 risk items. A related risk deals with 'Changing Scope/Objectives', which also was relatively highly ranked in all four countries. These risk factors together can be interpreted to reflect the universal difficulty of stating and managing system requirements throughout the development process.

TABLE 20 The rankings of risk factors (Hong Kong = HKG, Finland = FIN)

Risk Items	**Nigeria**	**HKG**	**US**	**FIN**	**Composite of HKG, US and FIN**
Misunderstanding the requirements	1	7	2	6	3
Lack of effective development process/methodology	2			14	
Lack of required knowledge/skills in the project personnel	3	13	11	3	5
Lack of available skilled personnel	4				
Under funding of development	5				
Importation of foreign packages	6				
Lack of "people skills" in project leadership	7		10		
Unclear/misunderstood scope/objectives	8	9			
Changing scope/objectives	9	5	10	19	7
Energy supply	10				
Artificial deadlines	11		7		
Inadequate user training	12				
Choosing the wrong development strategy	13				
Lack of top management commitment to the project	14	1	2	1	1
Failure to gain user commitment	15	3	4	8	2
IT awareness in the country	16				
Huge capital requirements	17				
Erratic and unreliable communication network	18				
Trying new development method / technology during important project	19		14		
Lack of adequate user involvement		2	6	11	4
Lack of frozen requirements		8	14	9	6
Introduction of new technology		12	12	13	8
Failure to manage end user expectations		9	7	23	9
Insufficient/inappropriate staffing		15	13	15	10
Conflicts between user departments		10	16	22	11
Lack of cooperation from users				4	
Change in ownership or senior management				5	
Staffing volatility			17	11	
Not managing change properly		3	4		
Lack of effective project management skills		5	1		
Lack of effective project management methodology		8			
Improper definition of roles and responsibilities		15	20		
Number of organizational units involved		17			
No planning or inadequate planning			5		
Multivendor projects complicate dependencies			12		
Bad estimation			18		
New and/or unfamiliar subject matter for both users and developers			16		
Poor or non-existent control			21		

'Failure to gain user commitment', however, was ranked much higher in earlier studies when compared with our study. This may reflect the differences in applications. In industrialized countries the expectations of the users may be higher and the applications built may be focusing more on adding value to user operations while in Nigeria the focus may still be on automating the basic clerical and business operations.

Finally the risk item 'Lack of Required Knowledge/Skills in the Project Personnel' can be seen to reflect the universal problem in managing and obtaining critical skills and knowledge to carry out the development task. The reasons for this can vary, however, considerably. For example, in Hong Kong a very high number reflected the high turnover between the organizations and the number of skilled workers who fled the colony before the Chinese turnover. In Nigeria this may reflect just the weaknesses in the educational system and the difficulty of obtaining and retaining a skilled work force in general.

Altogether, we agree with Schmidt et al. (2001) that the four panels perceived the relative ranking of the solicited risk factors differently. One explanation for this can be the level of control that the project managers felt they could exert over the observed risks (March & Shapira 1987). As March and Shapira (1987) suggest this is fundamental in understanding how managers actually approach risk. Based on the experienced level of control the software risks can be accordingly divided into the *outside risks* (over which the project manager has no control), and *inside risks* (which can be monitored and controlled). Between these two ends lies a middle ground of risks over which the project manager has limited, or shared control. TABLE 21 classifies accordingly all four lists using this classification.

By examining TABLE 21 by columns and tallying how the ranked risks were divided within each country into these columns we can learn the impact of the experienced level of control (or lack of it) on the risk identification and ranking. In this regard it seems typical of Nigeria to be that either risks over which project managers have no control or influence or risks with limited control by a project manager are prominent.

Whilst Schmidt et al. (2001) observed that outside risks were not widely selected for ranking, and those that were selected did not rank very highly, in Nigeria the number of outside risks was quite high, though their average ranking was not very high. Selections and rankings of items with *No control or influence* in each country are depicted in TABLE 22. In this group, Nigerian risks are more directed towards the environment over which the project managers had literally no control, while in other countries organizational level factors that are at least more predictable are more prominent. The total number of different risks under no control was 10.

Schmidt et al. (2001) observed that participants had a tendency to select and rank highly risks over which they had limited control or influence. This holds true also in Nigeria. As TABLE 23 shows the number of items with limited control is the largest and their average ranking is highest. The total number of different risks here was 14.

TABLE 21 Comparison of software risk factors selection (applied from Schmidt et al. 2001, p. 23)

Risks Outside Purview of Project Manager		**Risks Within Purview of Project Manager**
No control or influence	**Limited control or influence**	**Complete Control**
Erratic and unreliable communication network N	Under Funding of Development N	Misunderstanding the Requirements UHFN
Huge capital requirements N	Artificial Deadlines FN	Lack of Effective Development Process/Methodology HN
Importation of foreign packages N	Lack of Required Knowledge/Skills in the Project Personnel UHFN	Unclear/Misunderstood Scope/Objectives UN
IT awareness in the country N	Lack of Available Skilled Personnel N	Choosing the Wrong Development Strategy N
Conflicts between user departments UHF	Changing Scope/Objectives UHFN	Trying New Development Method / Technology During Important Project FN
Change in ownership or senior management H	Lack of Top Management Commitment to the Project UHFN	Lack of "People Skills" in Project Leadership FN
Staffing volatility HF	Inadequate user training N	Failure to manage end user expectations UHF
Number of organizational units involved U	Failure to gain user commitment UHFN	Insufficient/inappropriate staffing UHF
Multi-vendor projects complicate dependencies F	Energy supply N	Not managing change properly UF
New and/or unfamiliar subject matter for both users and developers F	Lack of adequate user involvement UHF	Lack of effective project management skills UF
	Lack of frozen requirements UHF	Lack of effective project management methodology U
	Introduction of new technology UHF	No planning or inadequate planning F
	Lack of cooperation from users H	Bad estimation F
	Improper definition of roles and responsibilities UF	Poor or non-existent control F

Legend: U=USA, F= Finland, H= Hong Kong, N=Nigeria

TABLE 22 Risks with no control

Country	Risk items	Out of total	Average rank
USA	2	17	10
Finland	4	23	17
Hong Kong	3	15	8
Nigeria	4	19	13.5

TABLE 23 Risks with limited control

Country	Risk items	Out of total	Average rank
USA	8	17	8.3
Finland	9	23	6.5
Hong Kong	8	15	4.7
Nigeria	9	19	9

Finally, inside risk items were ranked lower on average than risks with limited control (Schmidt et al 2001) (see TABLE 24). This also held in Nigeria. There was some variation between the countries, in that Finland selected a largest number of such items while the U.S. panel ranked them higher than the other countries. The total number of different risks was 14.

TABLE 24 Risks with complete control

Country	Risk items	Out of total	Average rank
USA	7	17	6.7
Finland	10	23	11.6
Hong Kong	4	15	11.2
Nigeria	6	19	9.5

Explaining differences between rankings. If we seek to explain some observed differences between countries one applied (e.g. Schmidt et al. 2001, Anakwe et al. 1998) model is Hofstede's (1980, 1991) theory and dimensions of cultural difference. Hofstede's dimensions of cultural differences are: individualism - collectivism, power distance, uncertainty avoidance, and masculinity - femininity. Individualism vs. collectivism refers to the extent to which individual or group needs and interests dominate. Power distance refers to the extent to which different societies are accepting of authority or unequeal distribution of power. Uncertainty avoidance refers to the extent to which a society can deal with uncertainty or ambiguous situations, and masculinity vs. femininity refers to the extent to which values such as ambition and aggressiveness are emphasized over nurturing and interpersonal relationships.

According to these four dimensions, the studied countries differ markedly (TABLE 25) if we apply Hofstede's way to divide Africa to West, East and South. Nigeria in this classification is part of the West. So, in Hofstede's original model the whole of West Africa – an area equal in size and variety to Europe – is considered as one culture. Most countries in Africa are colonial creations and usually there are several traditional cultures even inside one country that may differ radically. Thus the generalization to West Africa would be too rough in most situations.

The Hofstede-type variables have limited value when trying to sensitise devices to cultural differences (Walsham 2001). The best ways to develop cultural understanding are firstly, to live in a particular country and work there, and secondly, to read extensively about the particular region or country, its history, geography, social and religious beliefs. According to Walsham (2001), the benefits of reading and reflection on culture apply to the indigenous as well as foreign people. Our analysis is based on the in-depth case studies conducted together with the risk study, as well as literature.

Outside risks were emphasized in Nigeria. Usually risks that cannot be controlled are not considered as risks (Marsh and Shapira 1987), but constraints. One interviewee made an interesting observation that it is extremely difficult to rank risks that are common to all software development – like most of the risk factors are – on a par with the special risk factors in Nigeria. For example, the socio-economic factors compared with e.g. changing requirements are on two completely different scales. He continued that '*one has to do with risks involved in any ISD wherever it takes place, and the other set is particular to third world countries*'. Many of these risks are such that there is very little that a project manager can do and they hardly ever depend on him or her. Instead, we are dealing with constraints that are so pervasive, and part of the Nigerian reality that Nigerians accept them as an integral part of life, for example with the bureaucracy. However, the software business in Nigeria is so young (Soriyan et al. 2002), that the technically under developed environment creates a major threat for software development. In that sense discussing socio-economic reasons rather than cultural reasons for explaining risk selection and ranking makes more sense.

TABLE 25 Cultural dimension scores by Hofstede (adapted from Hofstede 1980, 1991)

Countries	**Cultural Dimensions**			
	Power Distance	**Uncertainty Avoidance**	**Individualism**	**Masculinity**
Finland	33	59	63	26
Hong Kong	68	29	25	57
United States	40	46	91	62
West Africa	77	54	20	46
Median of 53 countries	62	70	38	50

6.2 Case descriptions and analysis

The following sections describe the cases that have been conducted in order to figure out how information systems are developed in practice in Nigerian software companies. The purpose was to obtain some in-sight information about the issue so that the risks collected in the Delphi study would be better understood. Namely, what is the reality behind the software project risk factors collected in Nigeria. The case studies were conducted by interviews (see Chapter 5). The purpose of the case studies was not to interpret ISD projects in terms of cultural issues, but more in terms of social, economic, and infrastructural factors.

Initially the case studies consisted of four cases in three software companies in Lagos. However, the final cases consist of two complete cases and two incomplete, which I have combined into one case since they are from the same company, even if they are from different departments. They remained incomplete since both of the interviewees changed their jobs and we could not continue with the interviews. With the others, we had two different interview sessions. So, the two complete interviews are from 1. a locally owned quite small software company developing tailored or customised applications, and 2. a mixed owned quite small software company developing a packaged product. The combined case is from 3. a large foreign owned software company customizing imported applications. The third case is presented mainly to give a comparison to the other cases. In addition to the software companies, we visited a few of their customers. These interviews are referred to when extending the case analysis.

The cases are presented and analysed by using the Activity Analysis and Development (ActAD) method as a framework. The method is based on the activity theory, and the basic unit of analysis is a work activity, an ISD activity in these cases. The background of the method is explained in more detail in Chapter 3, and Chapter 5 explained how the method was applied in this study. FIGURE 41 illustrates the legend of the figures in case descriptions.

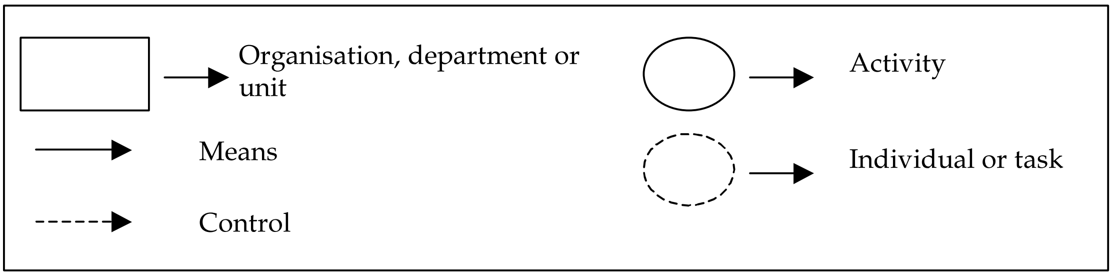


FIGURE 41 Legend of the figures used in case descriptions

6.2.1 Case of locally owned software company developing customized applications – Gamma Corporation

Background. The Gamma Corporation was established in year 1988. They had seven employees at that time. Nowadays its main activities are in IT training and software development. The company also provides some engineering and consultancy services. They have not specialized in any certain branch or application since the market in Nigeria is too small for that, according to the director interviewed. They have customers from government, banks, oil industry, parastatals, private sector, factories, including all kinds of big and small organizations.

In software solutions, the company applies two ways of selling software; outright package or license. The license means that they get a fee paid annually, but if they sell the package outright, they usually have a separate agreement for maintenance. It normally covers one year after installation. The maintenance can include some development (e.g. for efficiency), but not new modules.

The company now has 35–40 employees altogether in Lagos and Abuja (the new capital city). The organization is hierarchical, divided into software solutions, training, and consultancy units, as illustrated in FIGURE 42. The training unit provides all kinds of training concerning hardware, software (Oracle, Microsoft etc.), and business packages. Consultants usually work for big companies, like oil companies. Gamma can have a consultant for example in an oil company placed for a year. The unit of software solutions is taken for a closer look in this case. Besides the IT experts, there are also secretaries, some drivers and security men working for the company.

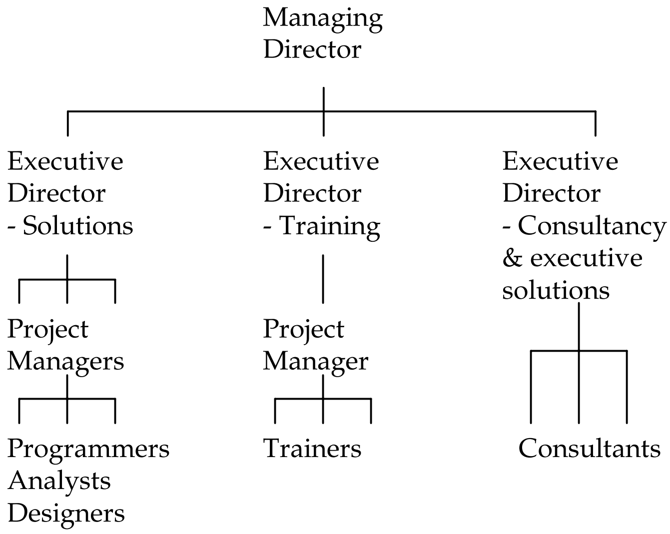


FIGURE 42 The organizational structure of the company

The managing director also works as the client services manager. He is not directly involved with the projects, but his job can be characterised like a '*political job*', when it comes to customer relationships. He also makes the company policies, strategies, and other general directions. Under him, there are executive directors, leading the departments. These directors form a management team of the company. The project managers (also called Assistance General Managers), who are responsible for the project teams in customer projects, are senior staff with long experience in IT business. They have the qualifications to understand the application area and functionality of it, and they have the knowledge about the technical environment. There are two project managers in software solutions. We interviewed the executive director of software solutions, as well as one project manager of solutions and one consultant. The consultant was a woman, the others were men.

This description, by using the case of the customer in the banking sector, highlights the software development activities of the company. The bank was quite big, having several branches all over Nigeria. Gamma has been in business with this customer earlier, providing training services. So the contacts had already been created before a software development project started. The bank was going through a business process of re-engineering (BPR) work, and needed a new system for human resource management (HRM) as a part of the re-engineering. The old system for HRM had been developed in-house some years earlier. Gamma had also developed a package for HRM, which had already been customized for a number of different customers. So Gamma conducted three demonstrations on three different levels at the bank; one for technical people, one for users, and one for the management. The executive director took care of the demonstrations. The bank had asked offers from several software companies, and finally they chose Gamma. This was during June-July in 1998, a year and a half earlier than the first interview.

FIGURE 43 The company main bulding

Object. The object of the project can be considered from two perspectives. Firstly, there was an activity of human resource management in the bank, which had been managed by using the in-house developed software application for several years now. Initially the activity wanted to be facilitated by modern information technology (e.g. the millennium was approaching), but the on-going BPR work was about to cause other changes as well. Secondly, there was a package for HRM developed by Gamma. The package is based on modules, like recruitment, hiring, job changes, addition of salaries, loans administration, pensions, payroll - everything that concerns human resources as used by a number of organizations. The package has been developed as user-friendly as possible with the tools available, but for being competitive, Gamma does not compete with the prices when it comes to their products.

The normal procedure was to *'go through'* the system with the client, in order to meet each customer's peculiar needs. For customisation the client might need for example new modules or changes to some reports. In this case, based on the old way of handling the activity, the customer had a broad list of requirements, which were at a very high level. With the help of that list, they went through Gamma's application and checked which modules meet closer to their requirements so that the amount of work will be less. Thus the first requirements were not very detailed.

According to the executive director, sustainability of computer based applications depends on proper training and local support, user participation in the development process (in terms of requirements), correct attitudes to computers, positive thinking, technical issues (correctness, reliability etc.), and an IT-department in the client site.

Collective actor. After the contract was signed by Gamma and the bank, they created a project organization. It consisted of a steering committee and an implementation team (FIGURE 44). The steering committee consisted of heads of relevant departments, head of audit and head of the IT-department (who was a woman) from the bank. The customer had an IT-department, who had developed the old system for HRM. Gamma's project manager attended the steering committee meetings as a consultant; he was not formally a member of the committee. The steering committee confirmed a project plan, which included a project schedule, all the tasks, activities, and resources. Overall, the steering committee had two main tasks: to solve financial problems and to take care of human resources in the project.

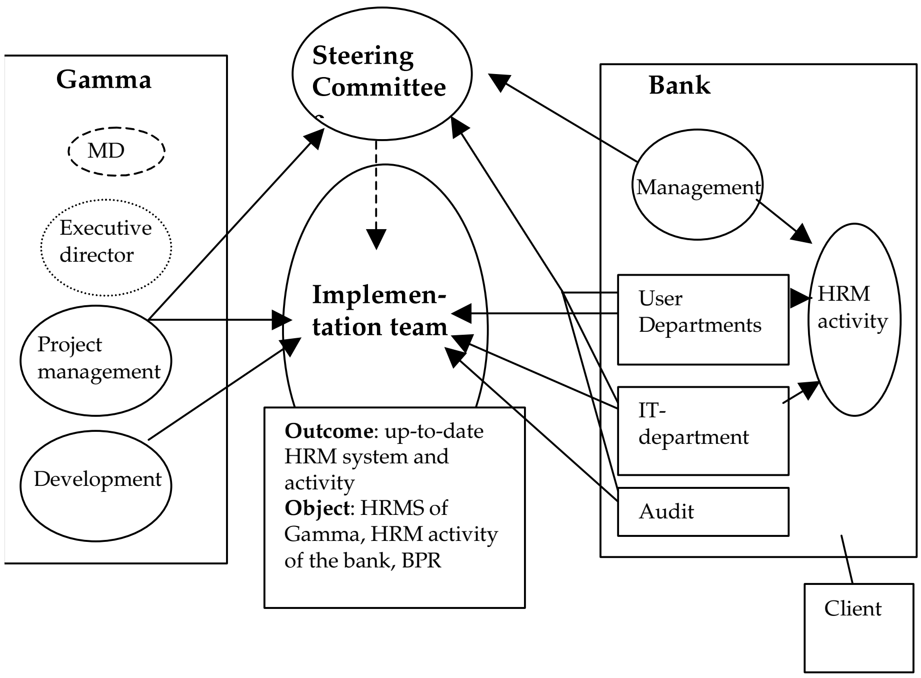


FIGURE 44 Project organization of HRMS development

The implementation team included people from both Gamma and the bank. From the bank there was a team leader (one of the senior users), users from different departments, the bank's own IT-people, and the audit representative. Since the bank's IT department had developed the earlier system, they had a very good understanding of the functionality of HRM and they provided the fundamental help in problem situations. From Gamma there were analysts and programmers.

Actors. In Gamma, 5-6 analysts or programmers were involved in the project at various times (two of them were women). All the technical staff had at least a Bachelor's degree in Computer Science, some had post-graduate degrees. They can also have some social-scientists or economists in different tasks. Since the company is also a training institution, they made use of the opportunity to train their staff on relevant courses in-house, and occasionally outside. The best training is to do the job: learning by doing. The new employees are usually young people because they are easier to train and they do not have many other engagements (e.g. family). The distribution between men and women is approximately 70 – 30 per cent. They also use students (trainers) for simple tasks.

They are considering offering Microsoft certification program to some of their own trainers since they are giving Microsoft training by themselves, and some of the customers expect that. As the executive manager said: "*the market is beginning to make noise about it* (certification)".

All the programmers start by only doing programming tasks and gaining experience. Some of them with the "*right skills*" are sometimes sent to clients and some of them also become analysts. The right skills mean skills to discuss with customers and the knowledge of the product. The most experienced analysts ("*top level*") work with the project director in designing and in requirement analysis. These experienced analysts actually do most of the design. They take care of the "*top level talk*" with the clients, meaning that they discuss requirements with the clients and also analyse these requirements. They take the responsibility for transferring requirements into the technical design. If the question is about a new system, they make a business system design and a technical design. The programmers – who are not that experienced in the application area – handle the programming. FIGURE 45 illustrates the composition of the implementation team. (The executive manager's expressions in quatation marks)

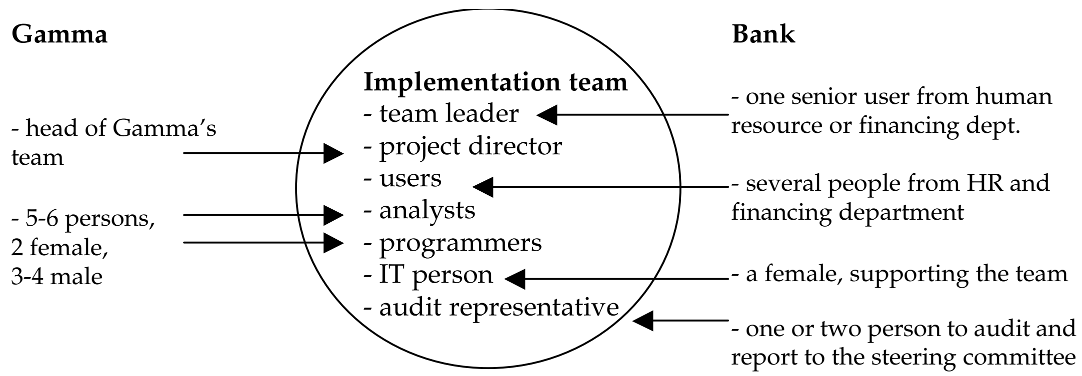


FIGURE 45 The composition of the implementation team

According to the executive director, it is important that the people in the implementation team have the "*right skills*". It means that Gamma expects users to be experts when it comes to their work. The customers expected that people from Gamma are experts when it comes to technology and application. Both the developers and users should be cooperative, capable of communicating with each other. The customers should also be willing to learn new technology. Sometimes the users are sent for Microsoft training to learn about the windows environment. However, the expectations are not always met. The team leader's role is very crucial and his attitude sets an example to other users. FIGURE 46 illustrates the composition of the steering committee.

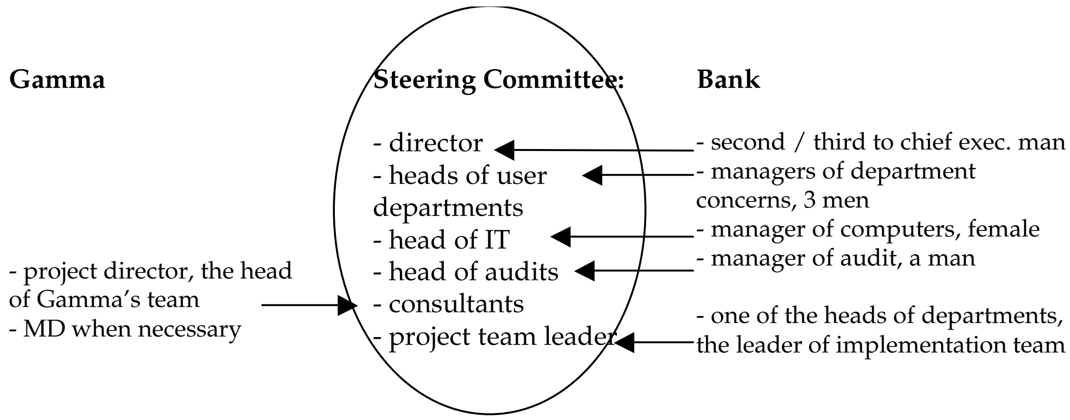


FIGURE 46 The composition of the steering committee

The company is very careful with customer contacts. They emphasize that "*it is different skills to talk to the user, listen to them, and build design than do programming*". The company wants to send only the most skilful and experienced analysts and programmers to meet customers. According to them "*client contact programmer must have an attitude*". In addition, they consider it very important to take users within the process, so that the users know what is going on and commit to the process.

Work process. Usually projects are supposed to begin with a *requirement analysis*, producing a system design report and an implementation plan, but in this case the requirements were initially based on the old system. They went through the application features with the customer (and users) and identified the needed changes for customisation. However, in mid-stream, an unexpected new set of requirements was given as a result of the on-going BPR. The *development* and customisation of the HRMS was like a rapid prototyping done module by module. They started by converting their own system to the client's environment, with minor changes. The IT professionals at the bank were helpful as they knew the functionality of HRM and understood computerization work quite well.

The modules went through two *test* phases in the development environment. First there was a test done by the Gamma people, different than developers. The tests were based on requirements and the testers used certain test scripts. This test phase included a quality assurance test in terms of documentation, screen design, error messages etc. After that the users tested the modules. In that phase there emerged new requirements for some of the modules. After acceptance the modules were moved into the production environment and the implementation was started. The most demanding aspect in the implementation was to get the old data into the new system, because of the lack of compatibility between the old system and new system. Initially, the customer wanted to enter the data manually, enter *"fresh data"*, but the computer crashed in midstream, and they lost most of the data since there was no backup. Eventually, part of the data in the old system was transferred using scripts made by Gamma. The rest they had to enter manually into the system, like *"fill the gaps"*. After all, it was more effective and quicker that way. The correctness of the functionality of the system was checked by comparing reports to the entered data. The customer did not keep the old system up-to-date in parallel. (The executive manager's expressions in quatation marks)

During the first interview at the end of 1999, the system was in use, but the users were still in the process of getting the data into the system. In addition, since the BPR was still on-going, there was one module under construction. The situation hasn't changed much within a year. The second interview was done at the end of 2000, and the system was still in the implementation phase because of the new requirements. The process had taken longer than was initially expected. Accordingly, Gamma had given only quick training to the key staff that were involved in entering data into the environment.

Means of work. The technology used by Gamma is Oracle, Unix, and Java with Windows workstations. Medium and small customers use SysBase and Microsoft databases because of the cost factor. Software development is based on high-level tools like the Oracle Designer, Visual Basic, and JBuilder. In little projects, they do not always use case tools, but the design is made manually. The executive director wants to rely on intuition, which can be lost when using automatic tools. Standards for screens, security, on-line help, documentation, reports, etc. are more heavily used when they are developing from scratch – in customizing an existing package the tools guide the work. Database design is considered very crucial; they have very strong standards in that area. For requirement analysis they do not apply any special tool, they just interview users and develop a business design report. On project management methods they don't want to be too strict, just to keep the project on track, *"a few guys who come together and get things going"*. For technical documentation they have specialists who are responsible. The bank had some standards for the interface, security, integrity (company standards), and on-line help. FIGURE 47 illustrates the main means of work in systems development project.

In FIGURE 47, the sources of different means are marked outside the main box. For example, the standards are based on the regulations of both the client and Gamma. The steering committee also affects the used standards, especially in project management. The needed skills vary from technical skills and programming to logical thinking and business knowledge. The executive manager considered logical thinking and business knowledge more important, because for example an analyst must be able to understand a customer's work and problems, and analyse how to improve it with IT. Communication skills are also valued. Only a little part of the needed skills is provided by universities or other education institutions, most of the skills must be learned from experience. Part of the knowledge is acquired from local or international conferences and exhibitions. Management also go abroad from time to time to attend courses in order to obtain the latest knowledge. Also professional magazines offer information of the latest trends. Technological means and methods are mainly bought from local vendors.

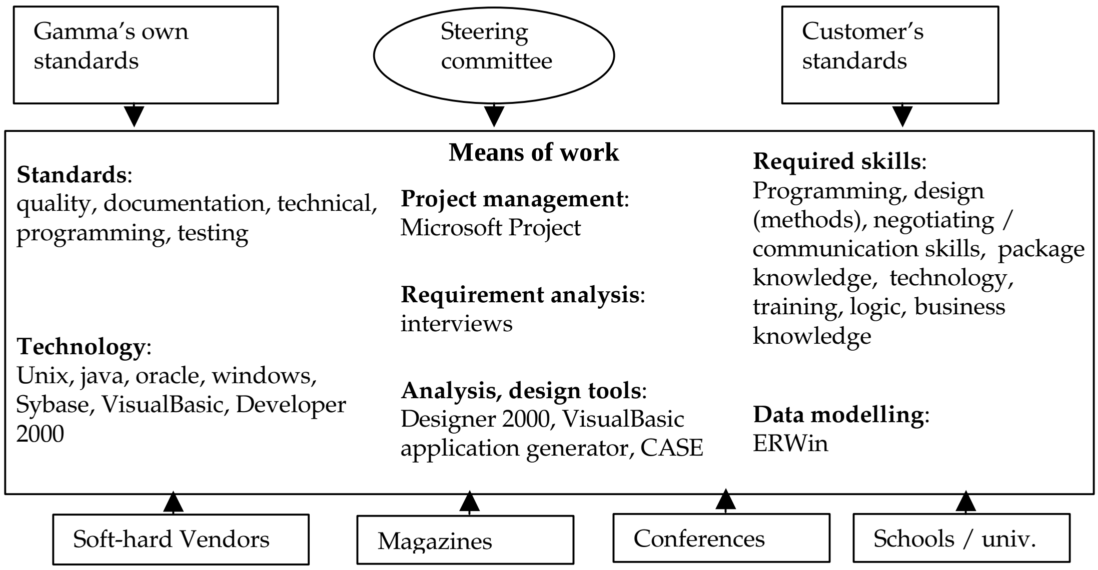


FIGURE 47 The means of work

Means of coordination and communication. Communication and coordination in the project was handled very much by meetings. The project plan, which was made by Gamma, was the main document driving the process. The communication and coordination within activities and the means of networking between activities are presented in FIGURE 48.

The steering committee had meetings once a month, sometimes every two weeks. In their meetings they checked the project situation against the project plan. The project team leader was the official link between the steering committee and the implementation team. The consultant, who was the project leader in Gamma, was a link between the steering committee and Gamma. The steering committee could also ask more team members to come in, or they could ask the consultant to leave. Sometimes the managing director of Gamma attended the meetings. Gamma's project management reported the progress to the steering committee monthly.

The implementation team met regularly, approximately once a week formally. In case of a critical situation there might be sub-meetings, not for the whole group but the people concerned. Besides meetings, design and programming documentation distributed the knowledge among the group.

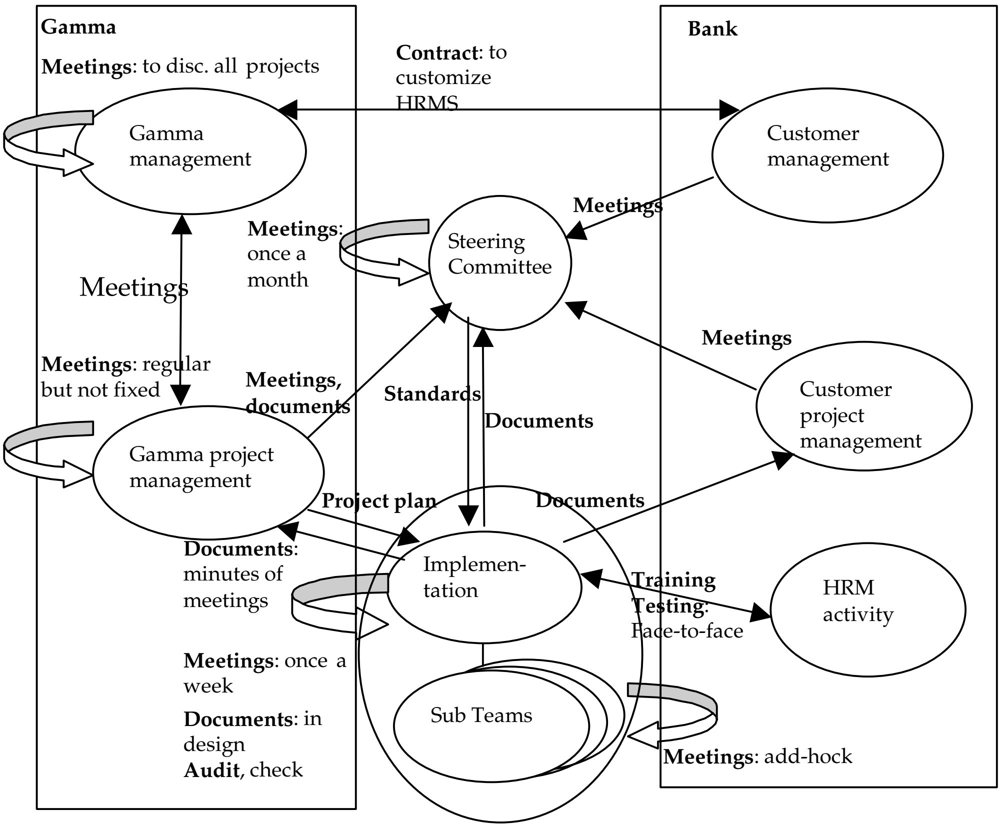


FIGURE 48 The means of networking *between* activities, and the means of coordination and communication *within* activities

Gamma's programmers in the project also met with users in testing and training situations. Usually a person who had developed a module took care of the training since he/she knew the situation. Only in special training Gamma used its training unit.

There were some checking points, "*crucial points*", when the audit representative came and audited the whole process in terms of requirements, security, and other standards. In this case the auditing was not very formal in the beginning. Later on during the process, the audit representative became more active and began to require some standards which had to be changed.

Usually, after the whole process, when the application has been installed and the users have been trained properly, the company tempts to leave the customer on their own, but sometimes it is not possible, and Gamma wants to "*make a customer happy*". If a customer is in a computerization process for the first time, it takes longer to leave them. For example, Gamma has one customer who at the end of the month runs payroll, and they insist that someone from Gamma has to be there. (The executive manager's expressions in quatation marks)

Mode of operation. The project was organized to have a very clear structure. Even if the operation was quite hierarchical, it was very cooperative. The key persons met regularly and also informally in order to keep things going. The whole package was divided into modules so that the process of implementation and customisation was easier to control.

FIGURE 49 presents a brief summary of the basic elements of the key activities, and the 'producers' of the means for the activities.

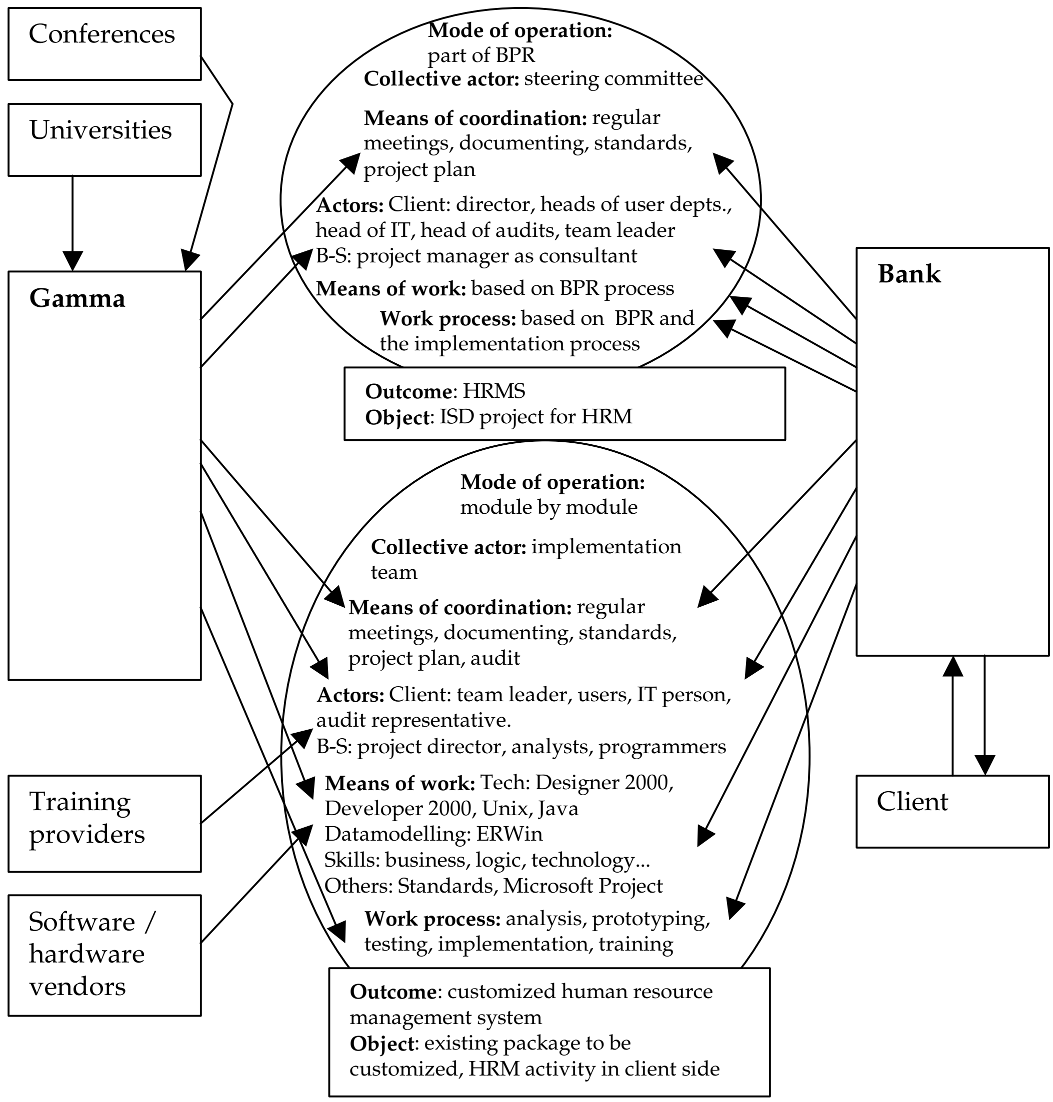


FIGURE 49 The elements of activity of the implementation team

The main problems. The main problems during the project were partly project-dependent, partly more general in nature. In the following we describe the problems that came out during the discussions with the company by using the same framework as in the description above.

Concerning the **object** of the project, human research management, problems are related to an old system and new requirements. The existing HRM system developed by Gamma was converted to a client's environment and customized to their initial needs. Thus the requirement analysis was based on the old system, even if the bank was in the process of a business process re-engineering (BPR). Actually, the HRM was the first issue on the list of re-engineering, and the bank was too eager to install a new system before a proper requirement analysis was done. That caused extra work during the process, since many modules had to be re-developed, and the final solutions of some modules were still under construction after long time. The old system as a basis for the requirement analysis seemed to be a mistake. The initial schedule was unrealistic and all the dead lines were passed.

The project organization (**collective actor**) was quite normal for the Gamma, with a steering committee and an implementation team, both headed by the customer. Usually the role of the steering committee is strong, but in this case the executive manager evaluated it as quite weak. Their impact was weak, but still they were the decision makers. The weak role of the steering committee can be caused partly because of the on-going BPR process in the bank and its total management, and partly because of the inexperience of the management to realize the embedded changes caused by a new information system. In addition, there were other stakeholders involved in the BPR, who affected the decision making of the client's management. One of them was an external consultant. Thus the situation was not very clear. Also the users were inexperienced to analyse their needs in the beginning of the process. This led to the situation of new requirements during the project. Gamma wanted to make the client happy and to get a good reference, so it wanted to be flexible. All the same, users were eager to use the new system, thus they had a positive attitude toward the process.

What comes to **actors**, concerning the analysts and programmers in the Gamma, the main problems in general were in their defective skills and knowledge. University education is lacking behind the requirements of the companies, when it comes to new methods and trends, but also, according to the executive director, in basic skills like database planning and logical thinking needed when designing and analysing object activities. It takes time for the company to train people for these important tasks. Gamma was (and is) very careful who to send to meet customers. Inexperience of the users for information technology can easily lead to attitude problems, but in this bank case, the problems were caused more by the inexperienced users to decide what they really wanted.

Problems concerning the **work process** were quite the same that has been discussed when analysing the problems with the object, focusing on the requirement analysis. New requirements had prolonged the process, and the plans and deadlines were not met. Also the audit representative became active in the middle of the process and started to require changes to different standards, for example in the interface. In addition, entering the data into the system faced problems with integrity and a machine crash, thus causing impatience among client's management. Also the training phase had not been started properly, because of the unfinished modules. Some of the key users resisted the start of the training, because, for them, training meant the last phase of the process, and they could not allow it even to start before all the modules were ready. The executive manager was sorry that the management level at the client side did not always really care for the contract, but tried to '*twist his arm*' and get more for the same price. The steering committee should have impacted on that issue. For now on, he had decided to be more formal and strict to what was agreed upon. However, he evaluated that the definition of the amount of work to be the most problematic issue in software projects. He was concerned that in the software business, time has a major impact on the cost.

The **means of work** concerning technology, standards, skills etc. were not that problematic in this case. The possible deficiencies of domain knowledge by the Gamma people were helped by the experience of the IT department people at the bank.

The **means of coordination and communication** were organised as usual, and there were no obvious problems. Still, the role of the steering committee was not strong enough to keep the coordination in hand, related to whole BPR process. The **mode of operation** was also organised as usual, with no obvious problems.

All in all, even if the problems seemed to be quite critical, the executive director did not consider the project as a failure. He emphasised that they have never abandoned a project, and they did not abandon it this time. Even if the project has not been that profitable, they have not closed any doors. This is their strategy to get clients and keep a good reputation, not force the prices down or using relationships. In addition, they have improved their work practices and knowledge about the domain, and this has been a valuable experience for some of the employees, who are better prepared for the next project.

Historical viewpoint. Gamma was established in 1988, having seven employees. The business idea was to provide IT consultation, which included various activities, like systems development, IT training, management services like providing server services, and providing IT personnel and computer equipment and installation services. Thus the activities included all the possible services clients needed when they started to computerise their own activities. The clients were big companies, three companies in the beginning in software solutions, including a bank and couple from oil industry. In Gamma they used personal computers (two computers), and the clients used mini computers. The consultancy work was normally done in the client site.

Quite soon in the beginning they developed the first version of the human research management (HRM) system. The first version was developed by Cobol and it was used by a few clients, for example one bank. Later on some of the clients wanted to have it on relational databases. Thus there was a shift from Cobol to databases in 1991. The change process was handled in-house. They set up a team, which was good at Cobol and knew about databases, and a committee, consisting of people who knew the system, to manage the change process. The change was not that big a challenge, because they already knew about databases, so the shift went quite smoothly.

By that time the number of personnel had risen from seven to about 30 people. They needed more employees since the number of clients had been increased to seven big companies. In addition, they still provided consultancy services quite a lot. There were not too many companies offering computer based services in 1990 (in 1988 it was 200 registered companies according to UNIDO 1989). The structural adjustment program (SAP) in Nigeria had increased the need for computerisation in the companies that survived (e.g.

Okuwoga 1990), and Gamma had already gained a good reputation. They also needed more staff who knew about the databases.

In 1991 the organization was much flatter. They had only executive directors and consultants. They usually had regular meetings on a monthly basis. However, the functions in teams were as it is now, having somebody as a project manager and others as programmers or so. It was more dynamic so that one person could act as a leader and as a programmer. In 'solutions' they had about 10 – 12 people at that time, so they had to '*network*' with each other quite smoothly. The phases of ISD consisted on requirement analysis with an outcome of a system design report (business rules, functions etc.), then after the approval of that, design, coding, testing, installation, and the produce of manuals (user, operation, technical), in brief.

The other major shift was the movement to an object-oriented client server environment in 1995, concerning the human resource management system. Again, their clients, especially one international organization, were asking for new versions, this time with graphical interface in a windows environment. Of course Gamma was aware of these features, but they had only one executive manager who knew about this new architecture and methods. He had to train other people for these, and it was a very challenging phase. They started to use Bower Builder for front end design, and SyBase as a database in windows 95, windows NT, and Netware environments.

This new shift did not increase the number of staff per se. They re-trained the existing staff. The number of clients has increased by eight new big customers. All in all, the technology had changed, but their activities around software development still had the same phases. It had not changed very much, since the development method and division to different activities has been created along the line during Gamma's history. Only formal standards and user participation got more emphasis among the graphical user interface. According to them, it has been extremely important to get real users involved in the requirement analysis, since for example, human resources had usually been handled manually in most places, and clients needed to adjust their old systems as well (manual activities). Gamma recognised that as a challenging part of implementation, which it still is. On the other hand, the users have become more familiar with computers, thus the cooperation has became easier.

After the shift to a client server environment Gamma has not faced big changes in its business. Competition in the software market in Nigeria has been increased quite much after the establishment of Gamma. There are new firms coming into the business all the time. However, Gamma has a good reputation and good relationships with its clients, so it is not that worried about the competition, at least the local competition. The favour for imported software application among organisations is a competitive factor, and Gamma challenges the government to support the local software business. In addition, the business climate in the country in general affects a lot. The private sector have had a major role in computerising the country (see Chapter 4), and creating market for software companies like Gamma. If the business climate is not stable, companies wait for better times for IT investments.

6.2.2 Case of mixed owned software company producing packaged software – Beta Corporation

Background. In 1994 two technical people and some progressive investors saw a possible market for software development business in Nigeria and they established the Beta Corporation. The shareholding of the company is 35% Nigerian and 65% foreign. All investors understood that high-tech companies will not give immediate returns, so they invested for the long haul.

The company started with a very flat and informal company structure. One of the owners used a lot of time to lay down the foundation. In the very beginning they had only a secretary, a cleaning woman, and a security man (who was in the hired house already). Initially four programmers were hired and trained, from 120 applications. It took some time before these programmers were able to do proper programming.

Initially the company set out to write customized software for selected clients. However, as most of the demand was for accounting software, the company ended up writing mostly accounting packages rather than small niche applications that was initially intended. Realizing that it was not cost-effective to try to design an accounting package from scratch every time a new customer came along, the company decided to take a big step and start to develop packaged software. The development project started in early 1996 and the package has been sold since mid 1998. For the original design and initial assistance with the accounting issues, the company was able to enlist the help of a brilliant accountant that designed a simple, yet powerful core for the accounting software. The development work was mainly done in a pilot project in one customer company.

The organization of the company is dynamic. Managing director (MD) and general manager (GM) are more like titles (see FIGURE 45), based on functions rather than positions. What they do changes over time according to the needs in business. However, the MD is the one who is responsible for business in general. At the time of the interviews, the MD managed also issues concerning technical implementation and production, and GM handled administration, marketing, and financial issues. MD, GM, and one active owner form together an Executive Board. The company has departments for sales, administration, accounts, and two real '*profit centres*'. The profit centres are concentrated on two main products: firstly, their own software (lets call it Beta-product), and secondly, the company is an agent for network associates, NAI, and for McAfee anti-virus application. The Beta-product development unit includes technical people, and people for support and training. Actually, the organizational structure presented in FIGURE 50 does not illustrate the grouping of people, rather the different functions and tasks.

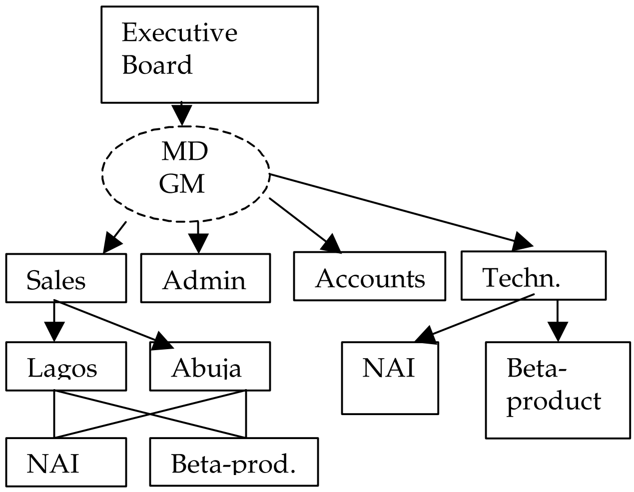


FIGURE 50 The organizational structure of Beta, based on functions.

The corporate culture is informal, with everyone calling each other by their first name. Although more structures have gradually evolved as per necessity, the organization is still flat. A flat informal structure relies to a large extent on individual responsibility and initiative, and great value is therefore placed on recruitment. The attraction of the company as an employer is not in high salaries, but because the company has a reputation for a dynamic, ambitious, and flexible company which provides its staff a place to learn and develop.

The company had about 60 installed sites in the year 1999, and the sales were increasing every quarter. Their customers consist of manufacturers, construction companies, some traders, one insurance company, and so. The users of the package are of three types:

1. Chief executives, who decide whether to buy a system or not. With the help of the system chief executives are not depend on the accountants that much any more. They can get information for decision making more easily.
2. Accountants, who are the ones that can be reluctant since their way of doing things might change. Thus accountants are important to *'bring in'* and take in training. They need to be assured that this is good for his/her career.
3. End users, who actually do the basic work with the system. If they are smart enough, they can do their job by computer in a very short time compared to manual system. In case they are very computer illiterate, the training must start with windows and other basic things.

The year 2000 was epochal in their business. In 1996 they ended up making an accounting product. In the year 2000 they decided to give up developing that product because of the increasing competition. They are not expanding the package anymore, they invest very limited amounts for it for now on. However, they are still selling it and they are supporting it, they have an intention to work with it for at least 3 years forward. In 2000, they got a new customer (government), to whom they were ordered to develop a taxation system. At the same time they had moved to a new technical architecture with use cases and UML modelling (Unified Modelling Language). Thus all of them had many new things to learn in 2000, including new architecture and the business domain.

The first version of the taxation system was done very carefully by doing the requirement specification with a customer formally. The activity was very formal, effective, and profitable. After this first pilot implementation the plan was to offer the application to the other clients as well, and form it into a product.

Anyway, we are now investigating their activity with the accounting package, the so-called Beta-product development. We interviewed the managing director of the company.

Object. The object of the work is a new version of the accounting product, which is based on an old version, and new development ideas, or bugs. The work is to make new features for the old system, fix some bugs, or to upgrade the system to a new platform. The competitive advantages of the package itself are price, local support, and user friendliness. In addition, the package is easy to implement and they have well trained implementers. The package is really a suite of applications that includes a general ledger, stock control, fixed assets, accounts receivable, accounts payable, cash, bank, payroll, human resource, and budgeting. Specialized modules like production control and insurance are also available. The package is a flexible system to operate by lots of options. It is made so that they can add features very easily to a particular company without having to change any of the underline code.

To be sustainable, according to the MD, the application must be user friendly and reliable, technically and practically. Thus the software must solve the problem at hand and the output must be reliable and useful. The cost of maintenance must be reasonable, also as minimal as possible, and further modification must be easy to do by the in-house personnel, as well as the support in general. The compatibility to other applications and platforms is crucial. In user organization, there must not be any negative 'side effects' caused by the application to current activity or other activities. For Beta, one key aspect is the reusability of code and design, they must be well documented. In addition, in order to get a good application development, there must be a good customer feedback media.

Collective actor. The Beta-product development consists of a group of analysts, designers, programmers, testers, and so on, including managers. In each task, the group of programmers is headed by the head of programmers. The development of the Beta-product is an in-house activity, without having any external participants involved, but the external stakeholders affect the development through contacts in implementation, supporting, and training activities.

To manage the implementation of so many sites, the company relies on external accounting firms (partnerships, not limited liability companies) that have been trained in the installation and running of the accounting package. In fact, a whole training and certification programme is in place to handle the certification of accountants to implement the package. The programme is similar to the Microsoft Certified Professional programme. Nowadays Beta charges for the certification. For marketing and selling the products, they also use partners. The structure of business is illustrated in FIGURE 51.

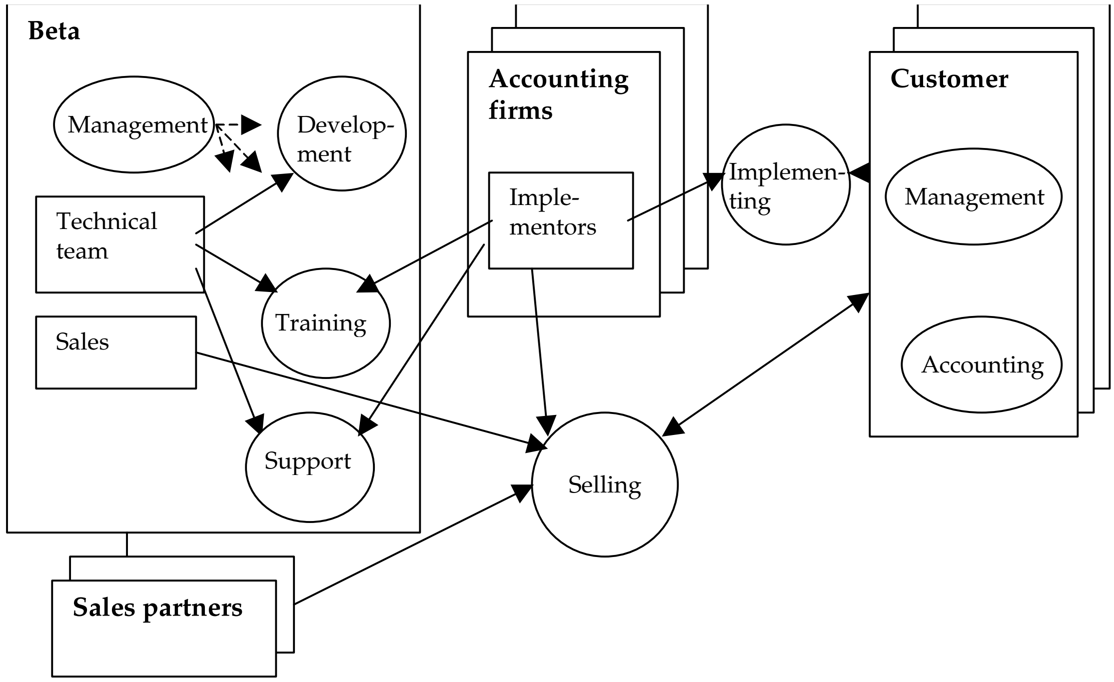


FIGURE 51 The collective system around Beta-product development activity

The main activities around the Beta-product are development, training of implementers, support activities for implementing and maintaining, selling the product, and finally, installing the product. For selling and marketing the product, the group of people participates if necessary. In this case, we have concentrated mainly on the development activity, but also taken a quick look at the implementation activity.

Actors. In 1999 the company employed around 40 people, and at the end of year 2000, the number was about 50 people, out of which 9 were active in software development. Other personnel include 6 technical people, 6 salesmen, and 2 trainers. The balance is managers, administration, drivers, messengers, cleaners, etc. Turnover of staff in the company is low. There are some people who have been in the company from the very beginning.

At the time of the second interview in 2000, they needed more people who knew about modern languages to rewrite programs, but in general, more crucial is the need for people who understand the business logic. They could have people with certain technical skills to do certain technical jobs, but they are dispensable. To have core people to understand the whole logic of the product and migration of the new technology is crucial. They worked hard to make existing people (who understand product) understand the new concepts and new architecture.

In recruiting new programming staff, more emphasis was laid on intelligence and brightness, than on technical knowledge and education. This is in keeping with the tradition of the company, where several people started more like hackers than computer scientists. According to the MD, graduates in computer science from Nigerian universities generally do not have the technical skills to directly enter production programming, nor are they up to date with the latest technologies, or even capable of designing relational databases. Ability to learn is therefore more valued than academic knowledge. This does not mean that the company frowns on skills. If people with the right skill sets were available, it is more than willing to recruit them. It would be quicker than having to train up people.

University people with a degree in computer science was a minority in the beginning, now they had some of them. According to a recruiting advertisement in the newspaper the company seeks people who have at least a degree in computer science, or some Microsoft exams (which are hard to pass), or three years experience in commercial software. For recruiting people the company has developed a rigorous testing method that included IQ tests, general computer knowledge, and test of specific programming skills. People to be tested are selected based on their CV. Sometimes the company recruits young people from the NYSC (National Youth Service Corps) camp. They go through the test and four or five people are selected for a year. After that contracts are considered again.

In additional training, traditionally the company relies heavily on the staff. They should have initiative and study by themselves using books, papers, and the internet. They have one basic text book in FoxPro (which they are using) to give everybody who came into the company to be read. The company encourages the staff to study and to do their MSc. Beta also provides some additional training, which is similar to all employees, no matter what their background education is. In addition, they give financial support to participate in Microsoft exams, as far as the employees are passing the exams.

Work process. In the beginning of the development process, Beta found it difficult to make a straight development plan with a timetable and resources. It was more like learning by doing at that time. "*Sometimes you go too deep into the water and you just have to learn to swim*."

Later on the activities became more structured, planned, and divided. However, the development of packaged software is different from a bespoke development. In packaged software, there is really no beginning or end. One version replaces another version, and minor revisions of this version are constantly released to fix bugs or add features. The development circle began with envisioning the new features or corrections, concluding requirement analysis, then there is planning of how to do it, then design and coding. The first release is given to users for testing. In accounting business the development cycle was quite long.

In summary, the development process is as the following, according to MD:

1. Envisioning
   - Deciding what features to include in the next release.
   - This can take place as brainstorming sessions.
   - Input of customers and implementers crucial at this stage.

2. Planning
   - Develop functional specifications.
   - Estimate time and resources.
   - This task is usually assigned to a responsible individual.
3. Developing
   - Design and coding.
   - Assign to one or more programmers.
   - Coordinated by the person who developed the functional specifications.
   - Also supervised by head of programming to check coding standards.
4. Stabilization
   - Bug fixes.
   - Testing compatibility with other modules.
   - Testing done by programmers and software testers.
   - At final stages, released to certain Beta Sites.
   - Bugs fixed by the programmers who wrote the code.

For envisioning, the group usually consists of sales people, support people, implementers, even top management, and head of programmers, who put the ideas into the design. Then the plans are put into production. In that group there are programmers who are headed by the head of the programmers. The development could be divided into sub groups.

Each new project is documented in a special format, which includes a user requirement specification, database schema, functional specification, documentation of public methods and objects, and an outline of any graphical element of the design.

While working on developing new features, there is also a constant necessity to upgrade the underlying platform to keep abreast of the changes in technology. Here a very different development strategy is needed than when it comes to adding more features to an existing platform.

While the company now has several people able to envision, design, and implement new business modules, the skills for changing the underlying platform and developing a structured plan to move the existing code to the new platform are scarce. The task of designing a blueprint for these structural changes falls at the present on one or two individuals, just as in the early days the business design fell on a few individuals. It is hoped that the skill sets for these types of architectural designs will be spread to more people.

Means of work. Concerning technology, the package runs on Windows and is developed on Foxpro 2.6, currently they are using Visual Foxpro 6, which is "*a very sophisticated data-centric object oriented language*". The latest technical challenge is to move to Windows COM objects. Parts which have nothing to do with data access, can be written in any language (VBM, C++ or so). The company usually buys its hardware tools from local vendors and software tools from abroad.

They do not have any legacy packages running any longer. Everybody who is now running a Beta system has the same version, so all the systems are upgraded.

For planning they are applying use cases nowadays. UML is applied for design, like sequence diagrams. For modelling the case tools are used. The project management model adopted is based on the MSF model of versioned releases. It is a cycle with four steps (see the work process above): envisioning, planning, developing, and stabilization. The project management model differs however from the MSF model in one important aspect. The business suite is structured as a series of smaller applications that are tied up with an API that allows different features and modules to seamlessly integrate without having to change the basic code. This means that several project cycles can take place at the same time with different modules. New modules can be added, or old ones updated in this way.

The bigger the business is, more standards and regulars are needed. The company has lots of standards nowadays, many of which have been developed inside the company. In the code level it is an on-going process. In the redesigning phase they try to be more strict than in the beginning. The policy is that there is less and less freedom how to do something. A code without a standard way of doing, even if it is without bugs, does not have enough quality. This is where the biggest problems are, since people are hard to force to use these rigid standards. Also in documentation strict standards are used.

Beta also has some self-developed support applications for work management, like version management application of modules (who did and when) and bug-checking application. The summary of the means of work is illustrated in FIGURE 52.

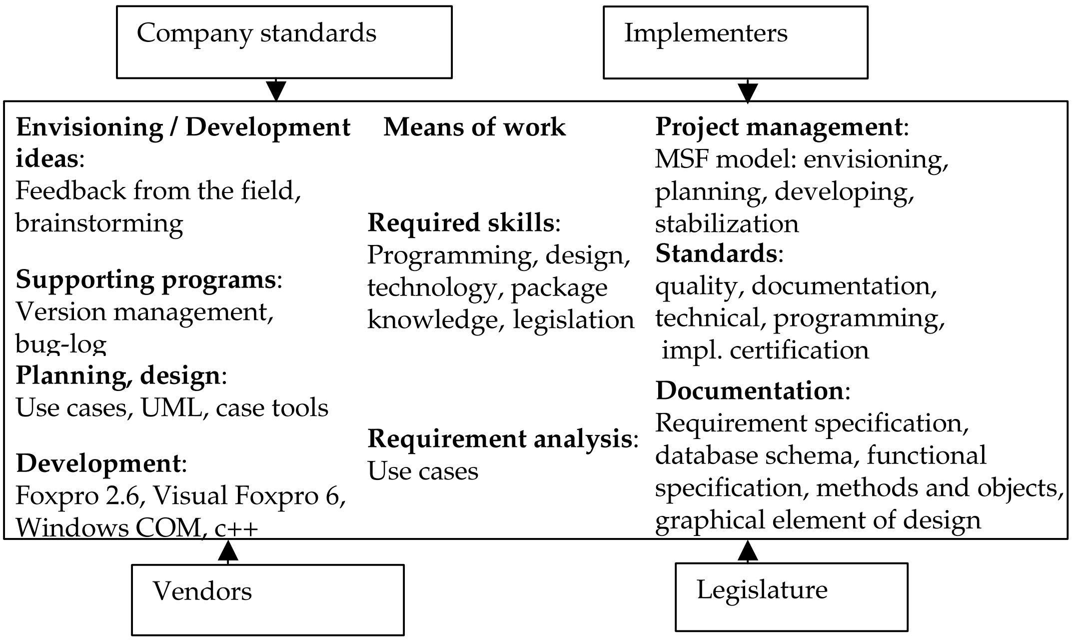


FIGURE 52 The means of work in SSL development

Means of coordination and communication. Concerning the means of coordination and communication the structure of the company is very flat and informal. People just come and go from room to room and discuss. Sometimes e-mail is also used, if for example the MD is abroad. The executive board has a regular weekly meeting. Within the new package production (taxation), the development group regularly meets once a weak, besides daily basis team working. All in all, the company has become more rigorous when it comes to a management of a development activity.

The collective part of communication is brainstorming sessions for envisioning. In that session all the stakeholders can bring new ideas or problems to discussion. Usually the ideas of implementers, clients or selling people go through supporting people. The implementer is also reporting to the production manager directly.

FIGURE 53 illustrates the network of communication and coordination between the main activities concerning Beta-product development, implementation between a customer and an implementer, and selling activity. The programmers do not meet users any more, even if they initially used to. Now the implementers handle the contacts. They give feedback from the field, usually through the support or training people, who are the same people and who also work with the technical team. After the implementation, a customer can, however, contact the Beta support people directly by telephone. In addition, Beta was thinking to organize a user group, but it never became very prominent. They had also put software on the internet, so that people could download the latest upgrade, but the problem with download from the web was that the internet was so slow. All in all, most of the communication with customers goes through the implementers.

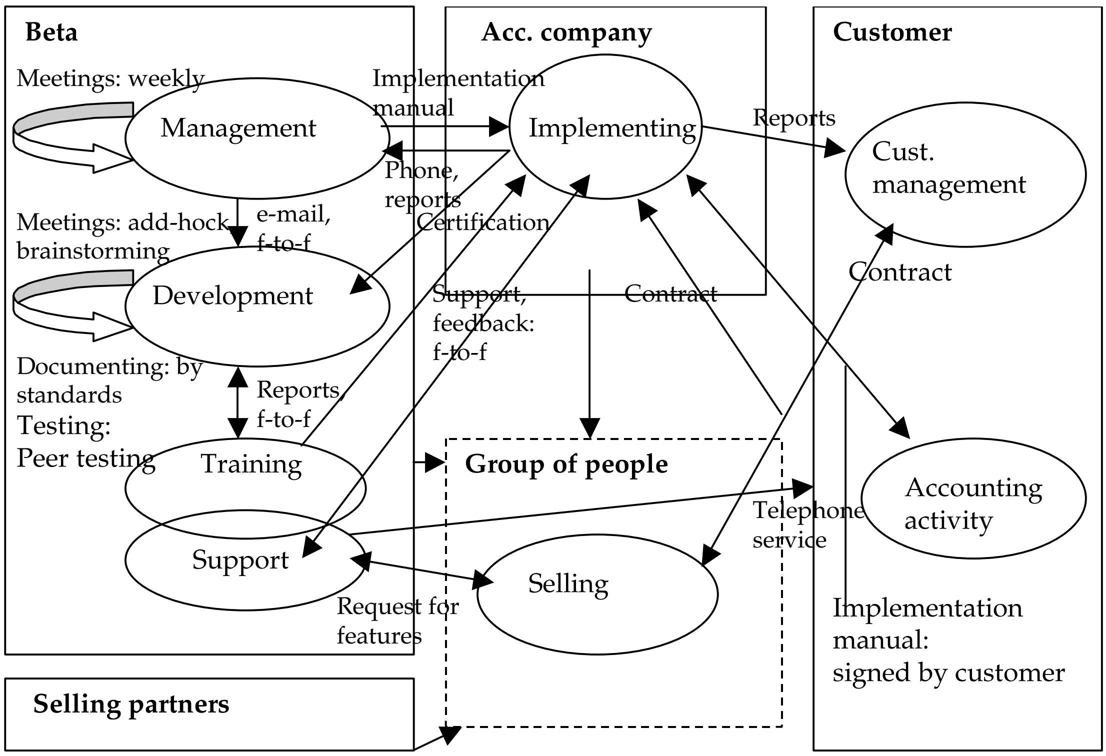


FIGURE 53 The means of coordination and communication within activities and between activities

Implementation is conducted following the implementation manual, which is signed by the customer. Implementers are trained and certified by the people from Beta. Usually the Beta management choose who is going to implement a new contract. The marketing and selling activity can be participated by many people, including Beta sellers, selling partners and implementers, when necessary.

As summarising, FIGURE 54 illustrates the basic elements of the development activity of the accounting product. The development takes place inside the technical team, only in the envisioning stage are there other stakeholders involved.

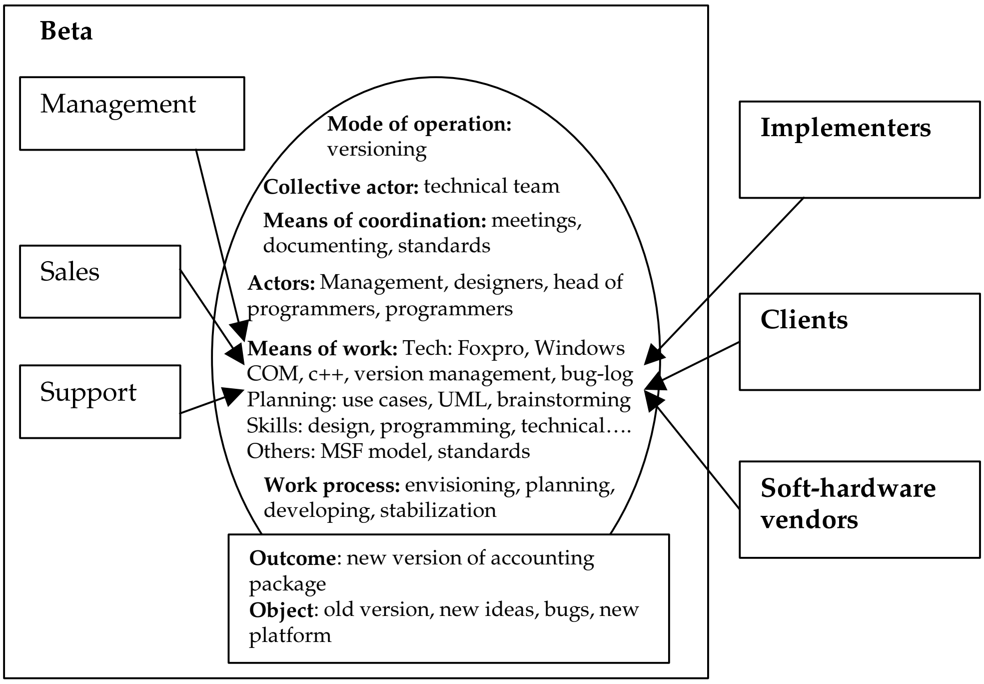


FIGURE 54 The basic elements of product development

The main problems. In developing the packaged product, the company faced several constraints. According to the managing director, the constraints can be summarized as follows:

- Lack of previous experience in the development of packaged products.
- Lack of staff with the requisite skills in programming, design, and project management.
- Lack of knowledge of business application software.

"Lack of previous experience in the development of packaged products. The founders of the company, though familiar with programming and computers, had no prior experience in packaged software development, or in large-scale development efforts. These two constraints made it initially difficult to create a structured project plan. Although there is much literature about software project management available, the implementation of such models is often difficult without prior experience and past data to use as a benchmark in planning. In addition, most of such models seem to be geared towards the development of a specific one-time application, and not towards the ongoing

development efforts needed in a packaged product that undergoes iterative versioned releases. As a result, it took several years for a working model for project management to develop, a model that is constantly being updated and revised."

"Lack of staff with the requisite skills in programming, design, and project management. The management of the company also faced the problem of not being able to recruit people with the needed skill sets. In the initial stages efforts were concentrated on developing programming skills in its employees, and design issues were by necessity left to a few people. Gradually some of the programmers acquired design skills, and were able to take up part of the design jobs. If people with the required project management skills had been available at an early stage, a formal structure could have been developed much earlier."

"Lack of knowledge of business application software. In addition to the other constraints, the knowledge of the company owners in business application software was limited. Business systems, such as integrated accounts, payroll, and production control systems, are often very extensive, and contain thousands of business rules that have to coexist. The primary information for the design of these systems came from customers and accountants. As few customers have the ability to understand all their needs when asked, it was impossible to acquire all the knowledge at once. The application therefore is developed in stages, often based on an incomplete understanding of the entire systems underlying the application. This means that the core design had to be revised on several occasions to accommodate new modules and new features in old modules. Sometimes when the desired changes were too drastic (i.e. involving too much work) to be practical to implement, the company would settle for a work-around that might be short of optimal."

All in all, Beta seems to have faced problems in its process to develop a packaged application, both in its management and employees skills. It took time and experiences to develop a mature project management method suitable for the company. However, they managed to do that. Thus, during the history the main problems have been in defining the *object* as a dynamic package which is correct and suitable for clients, meeting their *requirements*, to create a *work process* including a suitable management method to do business in the area of software applications, and to have educated and skilful *actors* for using the latest *means of work*, which are not familiar to just graduated students. Nowadays the last problem is still acute. In addition, the company had problems with illegal duplicates and software piracy, which is a familiar feature also in other developing countries (e.g. Heeks 1999a).

Historical viewpoint. When the company was established in 1994, there were about 500 registered computer companies in Nigeria at that time (Alabi 1994). Most of them offered hardware or training services, and most of the software was obtained from abroad. So there was a possible market for local software vendors. However, most of the needed applications were limited to routine activities, and also for Beta who mostly developed accounting systems. Usually accounting activities are among the first issues to be computerized, so there was a need for such systems. The first big change in the company's history was the shift from tailored software development to package development in 1996. Beta decided to take a big step and developed a packaged product for accounting, since it was much more profitable than to start every accounting project from scratch. Thus they decided to concentrate on that area. They also created a mode of operation to use external implementers to deal with the clients, since the skills needed in customer relationships and programming tasks are totally different. This mode seems to be very effective and practical.

The development of a packaged application was a learning process at the same time; how to develop packages, how to run such a business etc. (see problems above). The development of the company from 'hackers' team to a more hierarchical organization was quite evolutionary: '*learning by doing*'. Still, the company structure is still quite flat and informal, and it has been kept so intentionally. It gives employees more responsibility and independence, which is the strength of the company. On the other hand, they have gradually developed and increased strict standards, especially for programming and modeling.

The second big change during the history was faced in 2000, when the company changed the whole development architecture. It was a risky decision in the sense that part of the staff did not do any profitable job for a year, but developed that new architecture. Both management and employees had used a lot of time and effort to learn this new architecture, but the company was quick at reacting to changes in the technological environment. They were also quick at reacting to changes in the market. They recognize that the competition in the accounting software business was increasing, so they decided to gradually give up that business. In 2001 they had obtained a new possible business for the future, which arose from the agreement to develop a taxation application to the local government. At that time, they worked together with the client very closely in order to create a pilot application. But the use of implementers could be possible in due time.

The implementer viewpoint. We also interviewed one of the implementers, who had offered the services for Beta since 1998. He is the one who physically goes to the client site and takes care of the implementation process, after the contract has been signed. Thus the implementer is mediating between Beta and the client. The *object* of the work is quite challenging; get the old system (usually manual at the moment) and activity to fit with the new system, so that all the necessary information will be in effective use, and all the users will be calmed down to use the system. By the new application, the activities around the system face changes. Managers get information and reports faster and they can react more effectively. The most visible changes will be within the accounting activity, where the job descriptions can change totally. For example, earlier instead of three people, the payroll can be handled by one person, but the implementer has not seen any dramatic conclusions caused by the new system, since the people have been removed to other tasks.

Usually the implementer works alone with the customer. Only in problems concerning software, do the Beta technical people become involved.

The implementer is trained for the job on a two week course, provided by Beta. Usually a new implementer receives some chartered accountant as a supervisor, so that there are two people to manage the site in the first projects. It is like a practical experience for a beginner, but the accountant who was supervising our implementer won in a lottery and was away for 3-4 months. Thus the implementer was quite on his own in the beginning, and it was a difficult experience where he was forced to learn quickly.

The implementer had a degree in marketing. His professional background, started in 1990, was quite rich, including for example banking experience and working as a hardware supplier, and working in Kenya, but he has always had a strong wish to work on his own. He knew how to handle computers, and he had some accounting knowledge. That was enough, so he was asked to start as an implementer and he went on a two week implementation training programme to Port Harcout. Now the implementer can have several implementation jobs going-on at the same time, but Beta does not provide any basic salary, so all the incomes are from the agreements.

The implementer should have knowledge of accounting and computers, mainly how to use computers. He has to be patient and fitted with good self-confidence, since many times he has to take insults and not to allow that to harm the job. He has to be dynamic and sensitive. He has to control the job and be a little bit authoritative, as our interviewee described it. While the implementer works so close to the client, it is necessary to keep his distance, not to become too friendly, but to get straight to the point. Sometimes he can act like a consultant, for example by helping to contact hardware vendors.

The work can already start in the selling phase of the product. The implementer can be asked to go with the sales men to the client to demonstrate the application, since he knows the system as well as the accounting issues. If the customer becomes interested, they ask for a proposal for software and implementation. Usually the implementer makes an implementation schedule, which is part of the proposal (otherwise made by Beta). The usual implementation time is between six and eight weeks. Sometimes, if the job is complicated, Beta negotiates the price with the implementer, but if it is an easy job, they just name the price. The implementer gets 75% of the implementation charge.

After the proposal has been accepted and signed by the customer, the implementer, usually alone, but sometimes with somebody from Beta, goes to the client site and asks to see all records, books etc. He relates all the data to the structure of an application. According to him, it is crucial that the data is compatible with the application, otherwise there will be problems. For minor differences he tries to adjust the data, but if there is a need for changes in the software, the client negotiates with Beta. Anyway, after installing the software and entering data to the system, they print reports and compare the reports to a manual system. Usually he recommends that the client keep the manual system at least for a couple of months, just to confirm that everything is in order.

After the system has been installed, they start to train people. Usually the training takes two weeks. The implementer might put users to enter data into the system as a part of their training. They start do transactions and most difficulties are caused by the computer illiteracy of users. The implementer stays there until the users are able to use the system. After about eight weeks, Beta and the implementer leave the client on their own, which does not mean that they just disappear. The implementer goes there from time to time to check the situation. Sometimes the client can ask for extra training, which the implementer can do by oneself. Initially, the implementation contracts are made between Beta and the client, and Beta and the implementer, but after the implementation, the client can make a new contract for support, with Beta or the implementer. Many times they prefer the implementer because he is more familiar with the company. Beta provides an on-line telephone service for the users to help immediate problems. Otherwise, the client is responsible for the proper use of the system. For new revisions, the client contacts Beta.

All in all, the implementer is responsible for the implementation process, he controls the job. He does not meet Beta people if there is no special reason for that. He reports the process to the production manager of Beta and also to the client management. He also attends the meetings in the client site, when it concerns the application. Sometimes the customer may ask his opinion even concerning business issues. The implementer is a very crucial person to the success of a computerizing process in the company. He believes that in managing such projects, more importance is in experience than education.

The main threats for the successful implementation are caused by the software and people. Obtaining hardware is not a part of the implementation process. Beta advices the client about the kinds of computers they need. Software can be a problem, because it must be compatible with the previous system. People can form other problems. The reason is not just in computer illiterate users, but for example, sometimes corrupted management does not want the system to run correctly because it can thus harm their un-straight activities. Also people working in stocks can resist new systems because it makes stealing more difficult. According to the implementer, these kinds of situations are not rare, but if the system has been bought, it is usually based on serious thoughts, and the top management also wants to use it for the best.

FIGURE 55 illustrates the basic elements of the implementation activity. The mode of operation can be considered quite planned, fixed, and scheduled. On the other hand, it must be very dynamic and flexible, because of the unpredictable possible problems. The collective actor is a team, which varies according to the task within the application, e.g. payroll, invoicing etc. The implementer is a central actor. The means of coordination and communication is based on the implementation manual, but also the reports that the implementer writes to both management sites, but the basic mean is side-by-side working with the implementer and users. The means of work are records, books, and other sources of data. The just produced reports are compared to these data sources. The work process has been explained above. The important part of it is to ensure the correctness of the produced information of the new system. The outcome will be a new effective accounting system, which provides effective access of information for the management, and also causes changes to job descriptions of users and the accounting activity itself.

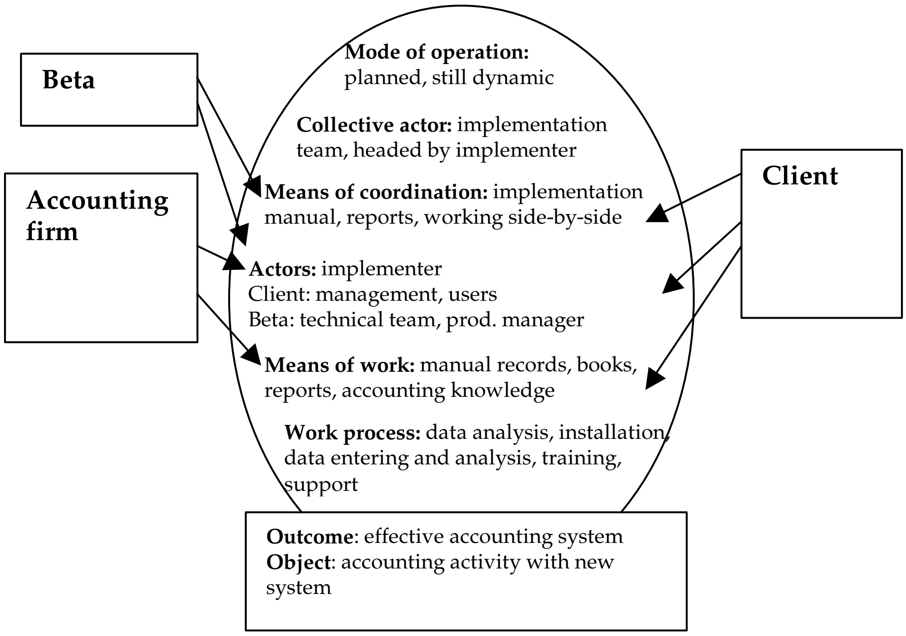


FIGURE 55 The basic elements of the implementation activity

The implementer believes that there is a need for local software providers in Nigeria. The strengths of the local software companies when compared to foreign companies lie in the local support and flexibility.

6.2.3 Case of foreign owned software company customizing imported software– Alpha Corporation

Background. The Alpha Corporation was established in 1982 as '*a solution provider*'. They sold PCs, provided design services, did programming, and so on. At that time they had about 20 people working in the company. Eventually they realized that there was already in existence solutions for different problems, mostly abroad. So they ended up providing 'ready made' solutions, meaning that they find programs abroad and customize these programs for local needs. Customizing means that they do not actually change the original source code, but they add some features like reports. The company's methodology is '*to find a good product, concentrate on it and use it to solve problems*'.

Nowadays they have about 120 employees all together. The company is divided into different departments: a financial business unit (FBU), an industrial business unit, communication unit, engineering unit, and technical unit. Communication and engineering units are to support business units. The technical unit handles hardware solutions. There is also a management team in the company, who controls all the projects. The international management control the business in general (FIGURE 56). We interviewed two of their project managers from financial and industrial units about two specific projects, one for a manufacturing customer and the other for a bank.

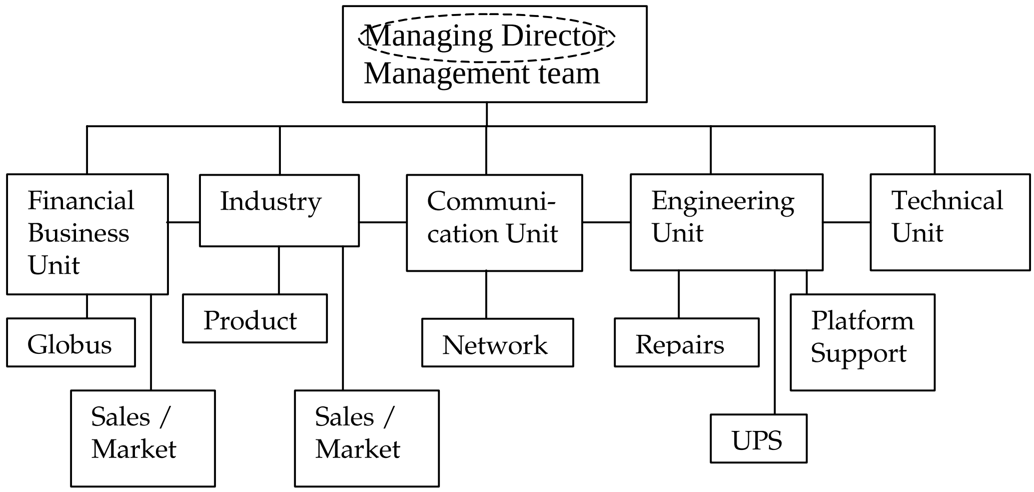


FIGURE 56 The company structure

Customers of the company are mostly big and / or international organizations, like banks and international industrial companies (e.g. Coca-Cola). The reason why the customers are only big companies is that little companies have too many other problems for example with a socio-economic environment and infrastructure that they can not afford the software implementations at the moment.

The organization structure of the company is very formal. The management team controls all the projects through a management representatives involved in every project. They report regularly to the management team.

Object. The specific projects concerned a manufacturing application, and a banking application. The manufacturing application was about to support the supplier - customer chain. The product was developed in 1970 abroad. It came to Nigeria in 1984 through this software company. The customer in this case received it in 1995. According to the project manager, the strengths of the product are in local support, the product is widely used all over the world and thus reliable, it is easy to implement, it is dynamic in its functionalities (database, interface, platform) and modules, and it is cheap to obtain and cheap to implement.

The application in the banking case, Globus, was developed initially in London and it has been marketed worldwide, also in Nigeria and South-Africa. The package includes all the aspects of banking activities. The product is structured as different modules, which are fully integrated, and which also have separate prices. Globus is a worldwide leader in banking software. It is quite simple to use, within two week the user can master the system, but it takes at least six months to know the product at consultant level.

Thus in both cases the software product was a well-known modular package developed in Europe for global markets.

Collective actor. In both cases the project was structured so that there was a steering committee on the client side. There was also a project team, which consists of people who own the process (experts of the process in question), users (senior users), and a project leader in the client side. From Alpha there were the project manager and some consultants involved in the implementation team. So, both of the projects were headed by the customer at a general level. Alpha's project leaders co-ordinated the activities of the software consultants.

In both cases, there were separate but partly overlapping teams in Alpha for project implementation and technical implementation. The project manager in the implementation team mostly negotiated and kept in contact with the customer and provided an undisturbed work environment for programmers and analysts. Usually the employees involved in the project need to know all the modules in the package, but in both units the idea was to train specialised people for each module, familiar with a technology as well as a software.

Besides the implementation or development teams, the units had two supporting activities: support team for training and taking care of fault calls, and demo or presentation team, which can also do some initial analysis during demonstration.

Thus the project organization in both cases was formal, headed by the customer site. The tasks as well as the responsibilities of each person were clearly divided and defined.

Actors. The people in the project consist of software experts and technical people, which they call consultants. They also have people who also have knowledge of the application and object domain, and who are in contact with the customers. In the banking project the technical team included 14 people (three women) in three sub-teams. The manufacturing project included a varying number of people in different phases; two persons at the final implementation phase.

The minimum educational background of people is a Bachelor’s degree, usually in computer science but also in other sciences. Most of the people come from Nigerian universities. When they recruit staff, they usually have a test and an oral interview. If a person has e.g. three years experience, he/she has to go to the test, but with ten years experience, there is no need to test.

The company values experience when they employ people only if it is from the same business branch. They do not expect earlier experience from young people, but for example in the financial unit, they insist on two years experience in the banking sector, before one can become a customer consultant.

Overall, it takes six months for an inexperienced employee to get the required knowledge in business and also in product, in general. All the employees are trained to know the product before they go into the field. The company provides inner training for example for customer care, team building, and accounting. If there is a need for an outside provider in further training of staff, the human resource unit of the company finds a suitable source. The company also publishes some magazines to spread information. The articles are many times business oriented, so that the readers are able to discuss with customers.

The usual age of employees is between 25 and 40 years old. For new employees they prefer young people, who are more dynamic in their working hours, but people who are consultants and meet customers should be old enough to be convincing. They have men and women, but the industrial environment usually is more familiar to men.

Employees attend conferences, not regularly but in terms of need or relevancy. They participate for example at technical conferences but also user group conferences, and more regularly COAN conferences. There are also workshops organized by other providers, where the employees can change their experiences concerning the application. The user group meetings have not been organized in Nigeria so far, but for example in South Africa.

Work process. The banking project started in June 1998 with a feasibility study, which took one month. Usually a feasibility study takes one week, but they wanted to make it good in order to have a good reference. Requirement analysis was conducted next. The financial unit had developed a picture of a system flow chart consisting of all the modules of the banking application. They started by going through the chart with the customer clarifying which processes can be used as a ready made package and which processes need more development. So they got a systems specification. They asked developers to evaluate how long it takes to develop such a service, and then they modified the specification according to the timetable and customer needs. They defined the programming language or tools, the number of man-power required etc. The work was divided into different teams according to what expertise was needed. After the development there was a test phase including a technical test and client test. The final test was documented to indicate that the test has been done satisfactorily. In implementation they fixed a performance month. If the system was not installed at that time the company received penalties, but usually they "*walk through it*" in time specified. If the project has some difficulties, they make changes to the schedule, otherwise the delays are not accepted. At the end, they started training in April 1999, and the implementation was finished in October 1999.

In the manufacturing project, after building the implementation team, they also studied the work environment. Alpha calls this as 'walk through method'. The package was implemented step by step, module by module. Early -96 they implemented the first outcome, which was 'financial aspect'. In April - 96 it was fully functionalised. In 1998 came the second aspect, which was the 'manufacturing aspect', and which was implemented in August -98. The whole process was completed in early January -99. Thus the implementation took about three months. What was good about the implementation project was the way they conducted the process. They followed the company 'guidebook', but partly they created and followed their own ideas, which were based on experience. All in all, in Alpha, projects should be well planned and documented, and followed. The company has a guidebook, the company standards, which provide straight orders and phases for each project. It is quite important to keep to standards, even if they allow some 'freedoms' for experienced project managers.

Means of work. In terms of technology, the packages run on different platforms, e.g. Unix, Novell, Win NT etc., with a relational database and Windows front-ends. The programming language is Progress in manufacturing system, and Universe RDMS in banking system. It is a universe Basic programming language. They use Unisys hardware, and the PCs are from South-Africa.

In the banking applications, the financial unit follows the Globus standard, whish is widely agreed among the partners around the world. All in all, the company has standards for software consultation and implementation. In project planning, the company uses Microsoft's Project for work plans, timetables, charts, and so on. When they have a new project coming, they use PNL, which is a profit -loss –analysis method. In installation projects, a project plan guides the process. It highlights what has to be done, when, and how. In documentation they have committed to ISO 9000-201. However, in customer projects, the documentation is usually adjusted to client's requirements. The legal departments of participants go through the documents carefully.

For analysis and design they have no scientific method. Normally the analysis in order to understand the problem is based on the 'walk through' process with a customer. FIGURE 57 illustrates the means of work in the company at a general level.

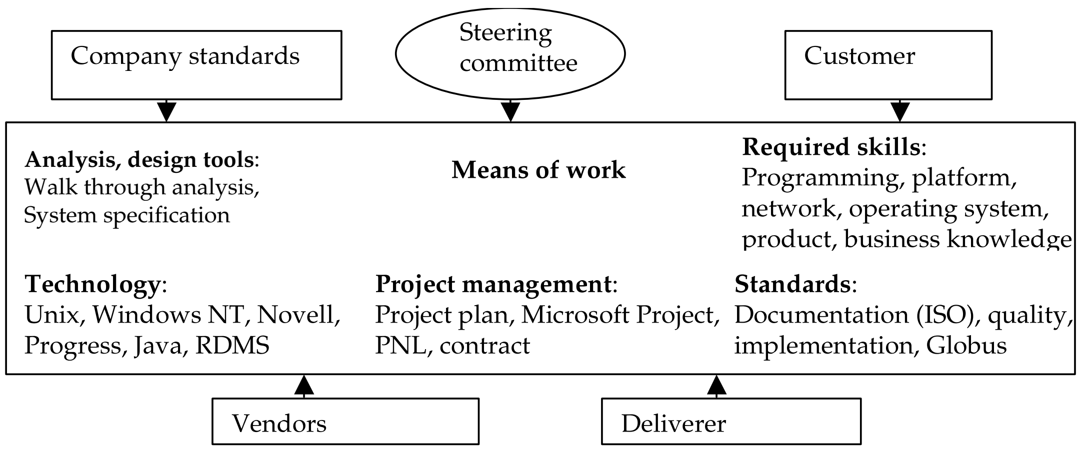


FIGURE 57 The means of work in Alpha

Means of communication and coordination. The meeting periods in the company are fixed. During the projects, "ground people" meet weekly. They have ad-hoc meetings as well if necessary. All the tasks are highlighted in a project plan. Reporting of projects seems to be a fundamental means of dividing information. Status reports are provided weekly or twice weekly if necessary. Each team reviews what the status of work is at the moment. Minutes of the meetings are copied for the managing director. Project managers participate in meetings in the customer site.

Management committee, which consists of managers and top managers, meet on a regular basis to review all projects – every fourth month. One member of the project team is also a member of the management committee. The managing director receives feedback formally and informally. Formal feedback comes from the minutes of meetings and reports. Informal feedback is mainly based on direct discussions with employees in different situations. At the beginning and at the end of the project there are formal ceremonies, where everybody involved in the project should participate. Sometimes they call the press, if there is a big project in question and they want to advertise.

In decision making, the project manager makes the everyday decisions and the manager representative implementation decisions and task distribution. Top management decides about facilities, tools, methods etc., but they usually consult the project managers for that. They can also delegate decision making for experienced managers.

User groups represent an important forum for customer relationships. For example in the banking application, Globus user group come together at least once a year to discuss problems and review the application. Each continent has its own user group. These meetings also provide an opportunity to meet the initial provider of the software. User training is also organized as a 'centralized' activity, according to a "*year schedule*". In acute problem situations, the units can connect to the client site by modem to check the situation, or they can go there personally. The clients keep a log-book of the error situations. Alpha also provides a help-desk to answer fault calls.

The other stakeholders of the company are international management of the company and the board. They also acquire knowledge of competitive market. Once in a while, they can for example send somebody to see the demonstration of some competitive banking application.

All in all, the means of communication and coordination follows the company standards and practices, without exceptions. The structure is very formal and static, but also effective. The company is responsible for the international management, which controls the activities.

The main problems. Even if the interviews with this company remained to a one discussion round, there are some problems defined, or risks in the activity the interviewees mentioned. The first mentioned problem was in user relationships in terms of requirement analysis and user involvement. The requirements can be over-ambitious. This has lead to delayed projects at high costs. Also the inadequate information of users' requirements is a risk. Sometimes end-users are not involved in project implementation.

The other problematic area was the unforeseen change of the business environment, especially for the users. Today the world can change very suddenly thus affecting a company's business direction. Accordingly the new and changed focus may invalidate the earlier users requirements already agreed.

In addition, the other project manager named a project planning as a problem. For example, really tight schedules or underestimated costs lead to unrealistic deadlines which are accepted by the implementation team. So, "*pragmatism should never be thrown to the wind*". One must also avoid "*cannot fail –syndrome*". The project should not be unprepared so that there are no proper plans for goals, requirements, targets, and so on. A project without direction is doomed to fail. So, in spite of the standards and regulations, the project planning can remain inadequate, or it can be poor.

A big problem is also the lack of knowledge and expertise of developers and implementers, in terms of technology or software. Part of this is the rush to new technology: quick to embark on new technology that is yet to be proved instead of old reliable ones. The other manger also complained of the lack of proper quality standards, i.e. knowing which international standard to use. Also the high staff turnover is harmful for a development of these skills.

The managers also mentioned the lack of top management commitment to have risen to be problems sometimes. The lack of real commitment from the top or rather feigned ignorance of incompetence to judge a project has lead to failures. In these cases, top management has played a passive role, which is risky. Also the lack of dedication, motivation or cooperation by team members is very risky.

Finally, there were some very practical problems. The lack of resources, in terms of tools, computers, manuals, money, training etc. is present in their every day business, even if they are a large and international company. In addition, the lack of proper management techniques or tools, or design methodologies was experienced as a problem by the mangers.

In the financial project in these cases, the biggest problem in the project was a technical one: java front-end of the client. They worked very hard and long to make the application work with this front-end language. They received help from the user group meeting, when they asked others experienced in such a situation.

All in all, the problems are surprisingly the same in this company when compared to the other two case companies, even if this company is large and international. Still, they faced problems with user relationships, especially in terms of requirement analysis, and also in project planning and the knowledge and skills level of their employees.

6.2.4 Extending the analysis

When we analysed the activity level of the case studies we conducted at these three software companies in Nigeria, their activities and most of the problems are not that different when compared to more industrialised countries. The problems faced during the process concentrated mainly to under specification of requirements and inexperienced users and their management. Similar problems can be identified in more industrialized countries as well (e.g. Keil et al. 1998). However, we can assume that these problems are on a little bit more of serious level in Nigeria, than the situation is for example in Finland. For instance, the requirement analysis is much more formal in most Finnish companies. In addition, users are nowadays much more used to computers in Finland than a few years ago. Thus they know what to expect from computers. This also concerns managers of user organizations. In Nigeria there can be situations where managers do not necessarily understand the difference between software and hardware: "*why should I pay so much for software, is it not enough if I buy a computer*?" Of course, this is not very common any more. However, our interviewees especially emphasised the difficulty to assess the time needed for a job and agreeing on that with a client. They also emphasised personal shortfalls and human relation problems that can risk the whole project, but these kinds of problems can be faced in Finland as well.

If we like to see the real contextual characteristics and problems of systems development in Nigeria, we have to extend our viewpoint to a more societal and economical level. The sources of the problems mentioned above cannot all be located inside the companies, they can be found in society as well. The history of software development in Nigeria is still quite short, so the society and the environment are not level to support the software business in the country. Several other studies in the literature (e.g. Waema 1996, Heeks 1999a) emphasise the problems in the socio-economic context of developing countries, related to information systems development and implementation.

The most visible problems that affect the business directly are *poor energy supply*, and *erratic and unreliable communication network*. Companies must put a lot of effort in to guarantee steady electricity generation; they need generators and stand-by generators. The lack of fuel forces them to buy fuel from the black market, which is extremely expensive. Unreliable communication networks blocks quick access of people and makes it difficult to arrange activities.

The *lack of resources for IT investments* is a problem of many organization in the society in question. Mainly big companies can afford to invest in software applications, e.g. banks, insurance companies, and oil companies. Smaller companies have to struggle to survive and they don't have enough resources for IT investments. Thus software business in Nigeria is dependent on the big and/or international companies for the moment. In addition, it is quite difficult to find new customers, since data about the domestic market is insufficient. The *absence of official IT policy* by the government (so far) has not helped the development of a local software industry. The IT investments have been incidental and the contracts are often based on relationships: '*man knows man*' is a Nigerian saying that influences business arrangements (c.f. Bada 2000). In addition, corrupted managers or officials, or employees inclined to misuse the property of the company, can resist new systems and even sabotage the development process.

The competition is hard because of *importation of foreign packages*. This is of course true all over the 'globalised' world, but for example in Nigeria it is difficult to compete with well-known foreign companies. Firstly, it is difficult to find people who know the latest technological trends since they are not taught at universities. The latest technology and books are also sometimes difficult to get, even for software companies, and they are expensive. As we have noticed for example in the Gamma case, the need for new technologies came from the client site. In addition, even if the first computers which went to Nigeria came in the 1960s, the history of indigenous software development is short. Like one of the directors we interviewed said "*we have no tradition in computer business*".

There have been cases where business ethics could have been better; the vendors had more or less a principle 'sell and run' (c.f. Korpela 1994). Thus many possible customers still prefer foreign rather than local software. Nowadays there is a registered association named Computer Professions of Nigeria (CPN), which is hoped to bring reliability to the business. Also CoAN (Computer Association of Nigeria) is promoting the reputation of local software vendors, but "*Rome was not built in a day*", as one of our interviewees said.

The society at large around the company matters. The *political and economical history* gives root to the enterprise. The embarkation of a Structural Adjustment Program (SAP) in the 1980s influenced the beginning of computerization in Nigeria (e.g. Okuwoga 1990), even if the program itself was not a success (Falola 1999). According to the software companies, being a dictatorship or a democracy has not directly affected the business, but in general the *environment* has its own meaning with e.g. overall safety and openness, possible riots or conflicts, general atmosphere, bureaucracy, taxation, and so on. For example, the more governmental the partner is, the more bureaucratic the activity is, and in some cases also un-straight.

All of the people we interviewed criticised that *university education* is concentrated too much on technical issues and programming, instead information systems (as social systems) and business thinking. Like one of our interviewee said, there must be 80% planning and 20% working in the systems development, but the university just graduated people who are too eager to rush into programming and then use 80% of work to do the work again. Thus, the bedrock of programming is the ability to think logically. In addition, the technical issues in the curriculum do not usually cover the latest methods or trends that are quite common in the industrialised countries.

One manager suggested that there should be one person at a university whose only job is to keep himself and his peers up-to-date on the latest technologies. In that way they could have an opportunity to follow the relevant issues that have to be emphasised in education. As he pointed out, everybody in the business should use 20% of their time learning. In addition, many of the people we interviewed thought that there should be more interaction and cooperation between universities and industry.

FIGURE 58 illustrates the societal and economical level that affects the activities of software companies in Nigeria, in quite a simplified way. The means for business is coming from abroad or local vendors, or they are in-house developed. NEPA should provide electricity and NITEL the communication network services. The money flows follow the provided means, even if the services are not always worth it. Governments in different levels (federal, local) are involved by giving legislation and by collecting taxes.

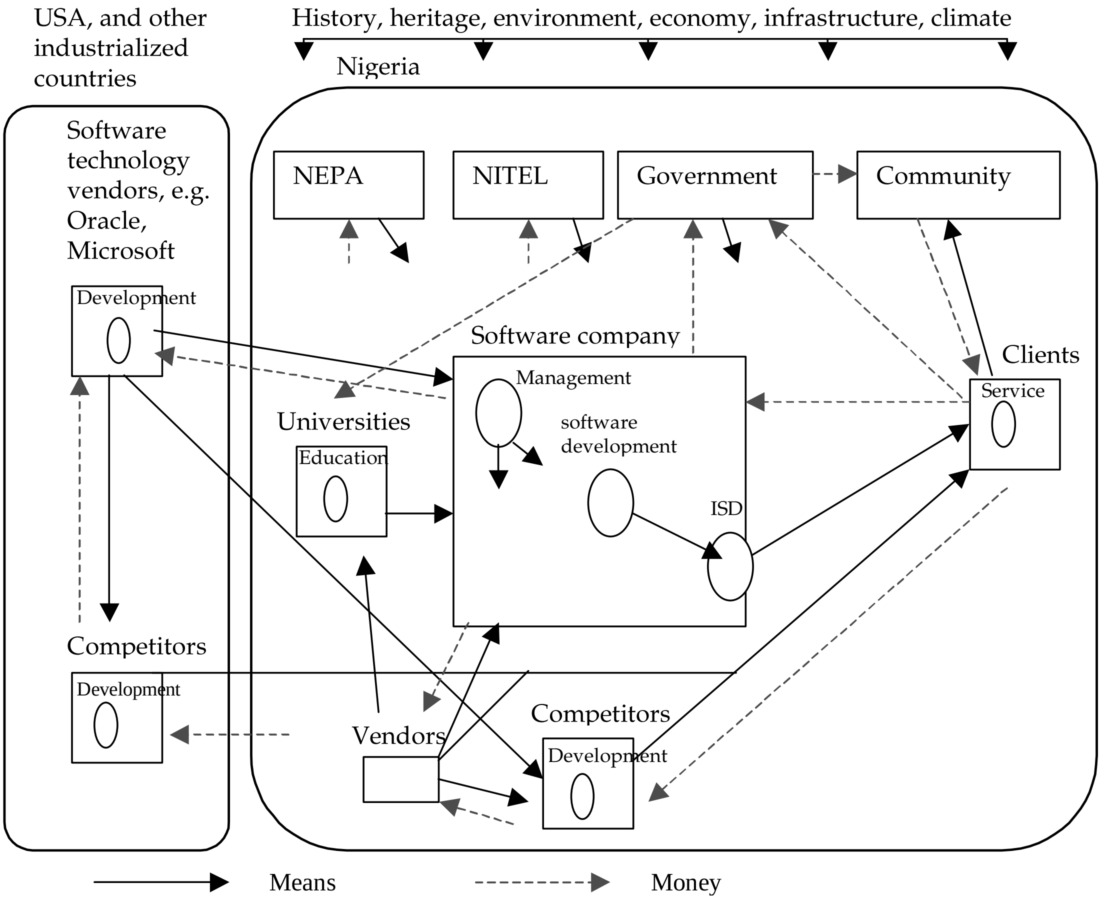


FIGURE 58 Societal level of analysis

Customer viewpoint. We visited two companies that had acquired a computerised system to replace a manual system in some of their activities. These companies were the clients of the first two case software companies, the other one was a manufacturing company (producing pots and pans) and the other one was an insurance company. Both companies were quite big and international, and in that way also advanced. It would have been ideal to visit some small, local companies with a computer system, but the software companies wanted us to visit these ones. However, I am not going to present these cases as I did with the software companies, but to give details that I consider relevant to the research, as additional information.

Both companies had more than 300 employees, and they considered that managing human resources properly is one of the most important issues to run a business. Thus they had both started to computerise their human resource management activity, first by acquiring a payroll system. The insurance company already had their first version in 1992, and now they had renewed the system by asking for the same software company to develop a new version for them with a Windows interface. Actually, the bank they were using to pay the salaries demanded from them to have all the information on discette. They did not accept manual lists any more (which the insurance company was able to provide with an old system). In 1992 they used a consultant to handle the implementation. The manufacturing company decided to start the whole computerisation with the accounting system, linked up to a payroll system. They wanted to computerise their activities step by step, not to take too big challenges at the very beginning. They were planning to go next to production in their computerising phases.

With the help of the new system they now had information for decision making more effectively and faster, and the information was more reliable. Thus the benefits of the system for the management were evident. The new system immediately highlighted errors and they received frequent reports of business issues. In the accounting departments the effects of the new system could be seen in the activity. Now the work that used to take three days can be done in a few hours, thus the system saved time and resources. Both of the companies said that the new system did not change the work procedure as such, not immediately. What it changed was a new way of thinking. For example, in the manufacturing company, the men in stock started to think the stock as their 'own business', since they now reported the situation daily. Before the reports were three months late, and the losses were big. So, perhaps eventually they will adjust their procedures as well, but the starting point of the new system was that it should follow their work procedure as much as possible. Both companies used a parallel to the old system to ensure that there will be no problems. Actually, the manufacturing company still used the manual system parallel.

The implementation in the insurance company was very quick and smooth, since they already had this old system. The software company had just converted the data into the new system. However, there were many new features to the new system that they were just getting familiar with, new kinds of reports and figures, and new ways of handling data, but they could use it as their old system. In the manufacturing company the process took more time than was planned, over a year from the time of the interview, but the reason was more to do with the busy manager of the company, not in the software company. This manager responsible for the process was too busy with other duties, so that some features were not yet totally decided how they should be. They noted that in order to improve the process they should have hired an expert for six months to take care of the process, an expert with proper skills. However, the systems were in use in both companies and the users and managers were happy with it. Actually, the users need not have any new skills for the new system. They were used to computers or learned to use it very smoothly, and they already knew the accounting issues.

Both organisations emphasised the locally developed software. For example, the payroll system in Nigeria differs that much form the British system, so software made in Britain would not be suitable. In addition, and what was even more important, they both wanted to have local support in implementation, and after-implementation support. It was emphasised how important it is to have somebody to quickly help them in problematic situations. The manufacturer had an IT department in their international level, but the software they provided was not suitable, and they ended up with a local vendor.

The system itself in the manufacturing company was running on the PCs, but they were planning for the networked system. The software company advised them to get appropriate technology, including a proper operating system. The software company sent specialists to install the network system. However, the manufacturing company had trained their own people to maintain the hardware. The manager said that what they need to maintain the computer system is "*money and patient*". In order to keep the system sustainable, they should have a "*reliable technology and trained people*". For now, they had an agreement with an external implementer (see case Beta above) to support the use of the new system.

In the insurance company, the system was running on the server in the accounting department. They were planning to have a network covering the whole company, but they still had some work to do with security issues. To maintain the hardware they had a contract with a vendor to prevent brake downs. In addition, they were very careful that the new system would interface with the other systems in the company. The company had an IT department (with four people), and there were people that they had educated for all the software in the company, but in bigger problems they could contact software vendors. They kept a log-book for every error situation. In order to keep their systems sustainable, they referred to the contingency plan.

For the manufacturing company, the computer system was not yet the most critical for production. They would survive in the case of the collapse of the system. It was not for making money, but saving money at that moment, but in future, the manager thought the situation will change in that sense. In the insurance company they considered the system more critical. In the break down, they could however rely on the spreadsheets, but that would be extremely slow.

In summary, both of the companies were advanced and serious in their computerising efforts. They both considered computer based systems to be important for business, in terms of reliable information and effectiveness. With a system they were able to control the issues that would not have been that straight otherwise. For sustainable use of their systems they emphasised the appropriate technology and trained people of their own, as well as local support. Finally, I like the reference to one of the managers in the insurance company when he said that "*it is very good that you* (foreign researchers) *come to visit here so that people abroad will see that we are doing serious business here in Nigeria*".

6.2.5 Summing up the results

It is now reasonable to compare the results of the risk study and case descriptions, as well as the results of the survey of *a typical software company in Nigeria* (see Chapter 4, also Soriyan et al. 2002) with each other. First, if we compare the three software companies with each other, the main differences were in size, user relationships, and in formality of activities. The locally owned small software company was less formalised in its activities and methods, relying on the tools they were using in documentation, and so on. However, it had the most closed relationships with customers. To be competitive against foreign competition, their strengths are in local support services, understanding the local requirements better, being able to customize, providing implementation support, and being flexible. To be competitive against other local software companies, Gamma relies on being flexible, and having a good reputation. The big foreign owned company was more formal, in its activities as well as in its user relationships. The clients of that big company represented the more international organizations. Perhaps it was more reliable in the eyes of the big clients because it was international itself, and large in size. In addition, they imported software from abroad to be customised locally. The mix-owned company which produced the packaged product had a little bit of a different business idea. The company had done a lot of work to make its activities profitable. However, it was quite informal inside the company, but more formal in its user relationships. They used external implementers to handle implementations. They seemed to have been very inventive when developing their business.

All in all, these three companies seemed to be quite different in every respect. However, they were all very professionally ambitious and advanced. The way they were conducting their projects is quite comparable with organisations in other countries, but when we consider their activity in terms of problems, the situation is different. It was a surprise how little the problems did differ when comparing for example the two customizing organizations to each other. For example, they both emphasised the difficulties in requirement analysis and in the knowledge and skills of their staff. The environmental and contextual problems were the same for all the companies, maybe less for the foreign owned company.

If we compare the results of the risk study to the case interviews, we can recognise many issues the interviewees also considered risky for software development projects. TABLE 26 illustrates the 19 most common risk factors identified in the risk study, and whether they were mentioned by the case companies.

In TABLE 26, the number of asterisks illustrate the importance that the companies put on each risk (* important, ** quite important, *** very important). These importance marks are based on the researcher's evaluations and intuition based on the interviews. So, what seems to be the common concerns by all the companies are 'Misunderstanding requirements', 'Lack of knowledge / skills of project personnel', 'Energy supply', 'IT awareness in the country', and 'Erratic and unreliable communication network'. If we try to find the reasons for these risk factors that all these different kinds of companies were discussing, we can simplify the source of risk factors to inexperienced users, poor educational level of universities, and poor infrastructure in the country.

The other risks were also present in the companies. Some of the risks can be caused by the missing IT policy in the country and the lack of government investments and interventions. These kinds of risks are for example 'Under funding of development', 'Importation of foreign packages', and 'Huge capital requirements'. Also the poor infrastructure and IT awareness in the country is a result of the government's ignorance for IT issues. The favour for imported packages can be understood by the short history of local software development, but the situation could be improved by the government, if they so wish.

TABLE 26 Comparison of risk factors to case descriptions

Risk Factors	Gamma	Beta	Alpha
Misunderstanding the Requirements	***	**	**
Lack of Effective Development Process/Methodology	*		*
Lack of Required Knowledge/Skills in the Project Personnel	***	***	**
Lack of Skilled Personnel	*	**	**
Under Funding of Development	*		
Importation of foreign packages		**	
Lack of "People Skills" in Project Leadership	**		
Unclear/Misunderstood Scope/Objectives	**		**
Changing Scope/Objectives			***
Energy supply	***	***	**
Artificial Deadlines	***		**
Inadequate user training	**		*
Choosing the Wrong Development Strategy		*	
Lack of Top Management Commitment to the Project	*		**
Failure to gain user commitment	**		**
IT awareness in the country	***	**	**
Huge capital requirements	**	*	*
Erratic and unreliable communication network	***	***	**
Trying New Development Method / Technology During Important Project		*	*

Legend: important *, quite important **, very important ***

The other risks can be considered to be more or less project specific and up to the project management, for example 'Artificial deadlines', 'Inadequate user training', 'Lack of effective development methodology to keep a project on track', 'Choosing wrong development strategy', or 'Lack of 'people skills' in project leadership'. These risks refer to the project management activity and echo the growing awareness of the importance of putting proper project management practices into place in Nigeria, and the risks associated with failing to do so (Keil et al. 1998).

The changing requirements and prolonged process in the projects refer to the results that indicate a lower level of alignment between software development and business development in Nigeria. This can be clearly noticed for example in the Gamma case. In addition, the software solutions have not been that critical to the business solutions to the same extent than in industrial countries. For example, when we consider the clients we visited, both argued that they could continue the business even in the case of a computer collapse. It would be difficult, but they would survive.

When comparing the cases to the survey of the Nigerian software industry (Chapter 4, Soriyan et al. 2002), we can estimate how good our case companies fit into the profile of a typical software company in Nigeria. In the sample of 103 software companies in Nigeria, 56% are also in Lagos, 89% are Nigerian owned, and the most of the companies have been established since the 1990s. In the light of these numbers, Gamma seems to be the most typical software company of our case companies. Foreign investors are not encouraged to establish business in Nigeria, due to the instability of economic and political system (Soriyan et al. 2002). Still, international companies with capital resources are less vulnerable for the instability, although the problems are the same when compared to local companies.

All our case companies seem to be a little bit bigger than that of an average software company in Nigeria (78% less than 15 employees), but the employees represent an average type, mostly men around their 30s, having at least a bachelor's degree from the university. In terms of customers our software companies seem to represent typical cases. The foreign owned company perhaps has bigger customers on average. What comes to services – imported packages (45%), local packages (28%), or tailored applications (26%) – our case companies represents all kinds. The technology they use is quite typical.

The projects in our cases were also quite typical, even if Gamma's project had been prolonged. That was because of the BPR work in the client site. However, the projects are usually phased, with around 5 people from the software company, and with a formal project structure. The project organization is typically headed by the software company, which was not the case in Gamma or Alpha. In that sense they did not represent a typical project organization. Their implementation teams and steering committee were headed by the customer, usually a senior user in the implementation team. In that sense these companies had realised the importance of user participation in order to obtain proper solutions. However, the used methods and practices in all the companies were typical (in-house developed, or object oriented). The result that adopted methodologies are mostly in-house developed partly indicates that the methods and techniques in books are not designed for small companies and too often they provide clean solutions to problems that are amenable for formal methods. However, the problems are usually 'messy' and less-amenable to be resolved with methods from books (Soriya et al. 2002).

The survey indicated that the companies are not satisfied with university education. The companies evaluated that their own ISD practices are in much better shape when compared to university education. Also our case companies needed to rely on internal training when they employed staff.

In summary, when comparing the results of the survey studies to the case companies, we can assume that the cases are representative enough from examples of the ISD projects in Nigerian software companies. On the other hand, we must consider that the companies selected for the study are probably more "advanced" than software companies in Nigeria on average, partly since they are members of the Computer Association of Nigeria and known to have some software development activities. Systems development practice in the smallest and most recently established software companies is thus probably not quite as professional, formal, and efficient. It must be noticed that the software industry in Nigeria cannot be separated from 'global' or foreign markets, since it brings pressures to the Nigerian software companies to improve their activities and methods. Also the education should follow the development in order to better serve the local industry, but the socio-economic and infrastructural problems and the collected risk factors are familiar to all companies – in many other developing countries as well (see e.g. Heeks 1996a) - and they cannot be underestimated.

It is obvious that the three different research efforts conducted in INDEHELA-Methods project did complete each other. The ability to interpret risk factors or the survey study becomes more reliable with the help of the cases, and the representativeness of the cases get a new light when they are compared to the survey or the risk study. By these three research methods we have obtained a pretty good comprehensive picture of ISD in Nigeria. When concerning the research questions of my study, the results provide data to software project risks, and methods, techniques, and practices in ISD. However, the question of the sustainability of IS use remained not clearly answered based on the empirical results; it is more speculative. We obtained some sustainability factors during the first round of the risk study, which are from IT experts' viewpoint. These sustainability factors are presented in Appendix 5.

7 DISCUSSION

The objective of this chapter is to discuss the results in the light of intended contributions. The obtained knowledge is discussed in terms of the research questions identified in the introduction. Based on the empirical data and literature, we can provide some theoretically argued requirements for ISD in Nigeria, and in Africa. To practitioners this study suggests methods for risk management and sustainability analysis. Also the used research methodology is discussed.

7.1 ISD in Nigeria

The research questions of this study concern issues that have not been studied earlier in developing countries or Africa. We wanted to know how the information systems are developed in Nigerian software companies in practice, what kind of problems they face and what kind of companies their clients are. We presented some research questions that we wanted to be able to answer after the study (see Introduction).

First, we investigated *software development* in Nigerian software companies. The risk study identified the risk factors that IT experts considered the most important in Nigeria. A surprise was how little the risks eventually differed when compared to studies conducted in industrialised countries. Surely there were some differences, but most of them concerned the socio-economic context of the country, and the infrastructure. These include a poor communication network and energy supply, and the uneasy political and economical situation. There were also some differences in the ranking phase, which have been discussed in Chapter 6. The most frequently named risk areas were the difficult socio-economic context, user relationships, and project management, requirement analysis, inadequate funding, and turnover of the IT personnel, and the lack of their knowledge and experience. The least attention received the corporate environment, scope, scheduling, development process, external dependencies, and planning.

It is difficult to estimate how much we can generalise our results throughout Africa, or to other developing countries. However, we believe that our results provide valuable insights into the variance of risk profiles in developing countries, which can also be utilized in other countries. The list of risk factors we collected provides a good overview of software development barriers. Actually, the list of 51 risk factors identified in this study has aroused considerable interest among practitioners in Nigeria. IT experts in the companies asked for permission to use it as a checklist in their projects. Also, the study made some project managers reflect upon their work, resulting in positive insight among the participants of the study.

The methods, techniques, and practices were clarified in case studies, as well as in the survey conducted by the project, which has been described in Chapter 4. The result was a notion that the methods, techniques, and practices did not differ much when compared to the industrialised countries. The tools and techniques were pretty much the same that are used in other countries, methods were mostly in-house developed. This refers to the issue that methods in books are not practicable in the small software companies that are most typical in Nigeria, but on the whole, the software companies knew about the latest trends and tools.

Secondly, *information systems development* was studied as a work practice in case studies, and the industry profile was clarified in the same survey mentioned above. Initially, we were not sure how many software companies there are in Nigeria. We had reason to believe that there would be some, since in 1994 there were more than 500 companies that provided computer-related services (Alabi 1994). By extrapolation we estimated that the figure would be 1000-1500 by the end of the century. We also estimated that there might be about 100-150 companies providing software development services. Based on the survey with 103 software companies we could increase our estimate to 150 – 200.

ISD as a work activity was studied by using the Activity Analysis and Development as a framework. We analysed the elements of the activity, like object, outcome, process, actors, collective actor, means of work, means of coordination and communication, and mode of operation. We also identified the problems that the software companies faced during the ISD projects. We noticed that when we kept the analysis on the activity level, the process description did not differ very much from the ISD work in industrialised countries. The problems they faced corresponded to the risks identified in Finland, the US or Hong Kong. These problems concerned requirement analysis and project planning, and the lack of employees with the right skills and knowledge. When we extended our analysis to a more social level, we found problems peculiar to that special context. These problems concerned inadequacies in infrastructure, education, and IT awareness in the country, including inadequacies of government investments and involvement. Thus the activity-related problems received new light as well, when we actually found the source of the problems.

All in all, when considering ISD in Nigeria, the work does not differ much from the industrial countries, when it comes to technology or software engineering. The main differences can be found from the socio-economic and infrastructural context of the country. Software companies in Nigeria have to struggle with issues that make it difficult to run a business. However, when we compare our results to the literature of information technology in developing countries, our research does not bring about new issues or surprises concerning the contextual problems in developing countries. On the contrary, our results support what has been written before.

What our results indicated and what is new knowledge is that *software companies in Nigeria are capable of providing the kind of software and information systems the organizations in Nigeria require*. At the moment, big customers seem to provide the local software industry in Nigeria with an opportunity to survive and develop, even if the former usually prefers imported packages, but the strength of the local software companies against foreign competition is in local support services, a better understanding of the local requirements, being able to customize, providing implementation support, and being flexible. The constraints in Nigeria indicated that *project management* should have an extremely critical role for successful ISD process, and it should be emphasised instead of the technical details.

Our results also indicated that in order to work for the development of the country – socio-economic or human development – software companies must realise that they are not living on an island. However effective and 'modern' they are, technological 'innovation' is not an isolated instance (Castells 1996). Castells (1996, p.37) writes that technological innovation

> *'reflects a given state of knowledge, a particular institutional and industrial environment, a certain availability of skills to define a technical problem and to solve it, and economic mentality to make such application cost-efficient, and a network of producers and users who can communicate their experiences cumulatively, learning by using and by doing.'*

He is talking about a '*milieu*' of exchange of ideas, problems, and solutions based on experience of the past. Maybe Castells refers more to 'Silicon Valley' types of environments, but this idea is also relevant in more 'modest' environments, like the software industry in Nigeria. The software industry in Nigeria is young, so it requires patience and vision to work for the development, not just for a quick profit. Thus the emphasis should be on sustainability of the would-be information system applications. It would also sustain the software companies in the future.

Based on the empirical results we can also evaluate the software business in Nigeria in terms of globalisation of the world. Software export is big business in some developing countries, for example in India (Heeks 1999b), as presented in Chapter 4. Many of the developing countries have cheap, talented labour, and software business is quite labour intensive, it has a relatively low entry barrier, and few economies of scale. However, the reality makes the situation more complex. For example, Indian export is mostly software services, instead of software packages, large amounts of the work takes place at the client's site in industrialised countries, and most work undertaken by developing countries is relatively low-skill software construction and testing (Heeks 1999b). Heeks also reminds the roadblocks, for example the infrastructure, which has been discussed in our cases as well. He also adds that many western clients still express fear, uncertainty, and doubt about contractors in developing countries, and the present players (like India) consolidate their position and squeeze out late-comers. Heeks sees five strategic positions for software companies in developing countries, as presented in FIGURE 59.

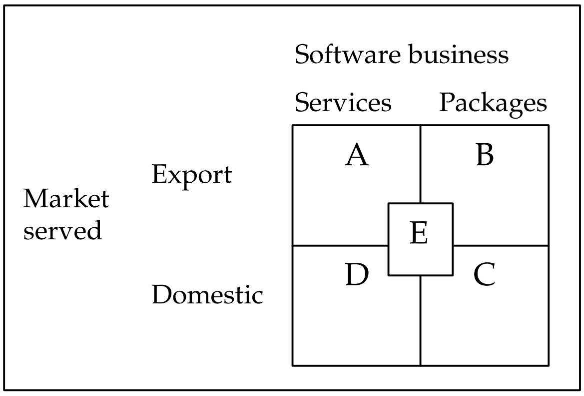


FIGURE 59 Strategic positioning for developing country software enterprises (Heeks 1999b, p. 16)

Many of the first-movers, like India, are in position A in a successful way, rather than in position B. However, position A is typically an export enclave in which skills and technology fail to trickle down to the domestic market. For example, India loses around 15% of its software workers every year, and the net earnings are far less than could be expected by the gross. Thus, this sector does not actually provide help for the domestic needs, and it does not provide a panacea for all. (Heeks 1999b)

In our cases and survey there were no software companies providing export services or packages as their major business. 24% of the companies in the survey had at least one customer outside Nigeria. We can presume that these customers were from neighbouring countries, like Ghana, or other African countries, since our case companies mentioned other African countries as prospects for extending business.

Heeks (1999b) sees position C as a difficult choice. It means competing with imported packages – legal or pirated. The low labour costs are quickly eroded by the package’s high development costs, and the preference for foreign packages, which our results also indicated, makes competition hard. We had one package-producing company in our cases, but also this company realised the increasing competition and decided to drop out of that business. However, they still had an idea of having a packaged product, but in a specialised niche market – taxation administration – what focuses on position E, discussed below.

The vast majority of software companies in developing countries are in position D, also in Nigeria; providing services for domestic market. In Nigeria some multinational IT companies also provide services, in collaboration with local partners, which indicates a pretty sizeable domestic market in the country, but the domestic market in many developing countries cannot be described as sizeable or demanding, that it can be considered as part of the global market (Heeks 1999b). This is the situation in Nigeria as well. As Heeks notes, most position-D companies therefore remain position-D companies: small fish in small pool. Position D is more of a survival strategy than a development strategy, but at least it is a starting point, and if the companies in position D are the best at providing solutions for domestic needs, it is a good starting point.

Going a little bit further, the domestic market can provide an opportunity with specialised niche markets, in position E, as has already been noticed with one of our case companies (Beta). Heeks separates the following opportunities:

- Sectoral niches like banking, insurance, health administration, hotel management, mining, and industry;
- Application niches such as Web browser add-ons or text retrieval utilities; and
- Linguistic niches for regional languages like Spanish, Swahili or Chinese. (Heeks 1999b, p. 18)

Actually, Alpha as a foreign owned company also includes this sector by providing banking applications and industry applications. We also met some small local companies, which had developed health administration applications for hospitals. We can summarise that in Nigeria the diversity of software companies in a domestic market is already notable, thus indicating a belief of software companies to a software business in the country, in spite of all the constraints. We can again refer to the competitive strengths of local software companies, mentioned above.

The third part of the research questions concerned *information systems in use*. The results for that part have been obtained from case studies, also from the customer visits we made, and the survey. According to the survey, the most typical customer of a software company is medium-size (51-250 employees) in private sector services (wholesale and retail trade, transportation, financial services, IT, and other business services, etc.). The customer companies we visited were in the category of large (251 – 5000 employees) with about 300 employees, in the insurance business and manufacturing sector. The typical organization that is using computers is quite big, and most times international. We can assume for example that the oil industry depends quite much nowadays on computer-based systems, but usually they mostly receive their applications from abroad, at least when it comes to production. Also other organizations mostly get their critical applications from abroad, but some companies have realised the benefits of using the services of local software companies, as our cases indicated.

Banks need banking applications, which are usually from abroad (c.f. Bada 2000). Banks also seem to have started to think about their business strategies, since one of our cases dealt with business process re-engineering in a bank, and such cases have been reported earlier as well (Bada 2000). Also the public sector seems to be a big customer, according to the survey. However, the IT investments of the public sector are incidental and dependent on the political will. After all, the organisations using computers and investing in IT must have resources, which are scarce for little companies. The tradition of '*man knows man*' business agreements will have less emphasise when the market increases. By the promised declaration of the national IT policy, the market of software companies could be stabilised, since the political environment effects other companies' investments as well. Thus the state intervention is very much welcome.

In our cases the typical applications were for accounting and human resource management activities. The companies were looking for effectiveness for activities, and reliable and fresh information. Having reliable information in time seemed to be one of the most valued issues for management. Both of the customers we met emphasised that they did not change their work procedures, only some practices became more fluent and efficient. They wanted the new system to follow their old working practices, so that the implementation and introduction of the new system would be smooth. They confirmed the sustainable use of the new system first by running a manual system parallel for some time, to make sure that there will be no problems. Secondly, they emphasised the appropriateness of technology, and the support services (for hardware and software), whether in-house or agreed with a vendor. Thus the sustainable use of IT is dependent on both technology and people. Organizational capacity building is thus crucial. The companies did not question the need of information systems, both of them considered it very important and even crucial for the future.

Grundey and Heeks (1998) identify '*natural protection*' to be one of the features that protect software capabilities, in Romania in their case. The natural protection is based on special user requirements in the country, derived from particular organizational practices and social, economic, political, and cultural environment. Grundey and Heeks found the following elements in Romania for protection: the Romanian language, the importance of personal contacts rather than mass marketing, and the process of transition and the continuous changes it demands. In addition, protection also derives from the high cost of imports in an impoverished economy. Also, to purchase larger foreign packages is a long-term investment, and in an uncertain economic and political environment it could be risky. Thus these protective factors delink local software consumption to some degree from the global market. Apart from the Romanian language, the above mentioned issues are valid in Nigeria as well. As in Romania, in Nigeria Microsoft is one of the beneficiaries of the growth in software consumption, but there are also plenty of opportunities created by local package customisation, training, support, and custom-built arenas.

The natural protection is not guaranteed or a permanent situation. It can be threatened for example by an adoption of work methods provided by imported packages in organizations (Grundey & Heeks 1998). Many organisations may think that modernising activities according to Western methods would be better than the 'old ways'. The clients we visited did not think like that, on the contrary, but it cannot be trusted that it will be so in many places. In addition, as Grundey and Heeks noticed in Romania, many multinational companies are localising and customising their packages to match the practices of individual country markets. This was the case in our case company of Alpha, but if they are using local counterparts, like Alpha, we can still talk about local services.

In summary, our study provides up-to-date information about information systems development in Nigerian software companies, the software companies themselves, and the clients of the software companies. The information can be valuable for the companies in Nigeria when they reflect on themselves in the business, and for the other researchers who are interested in developmental informatics.

7.2 Special requirements of ISD in Nigeria

The research approaches concerning ISD studies are characterized by diversity, focusing on different kinds of results, as noticed in Chapter 2. The idea of ISD as a universal phenomenon is less commonly questioned. We can ask if ISD really is the same in Africa as in industrialised countries. We can further ask if the very different socio-economic context in Africa result in some special requirements for systems development. By special requirement is meant issues that should be included in ISD methodologies and education. We have asked these questions in the INDEHELA-Methods project and tried to find some solutions or suggestions by conducting *theoretical analysis of special requirements of ISD in Nigeria and Africa*, based on the literature and empirical findings (Korpela et al. 2000b, Mursu et al. 2001).

As we have emphasised in our research, information systems are social systems rather than totally technical systems. ISs exist in a real world which consists of objects, people, rules, norms, and commands (Land 1994). Information technology is just a tool in a given work activity.

The ISD process is composed of activities and relations of different stakeholder groups such as users, IS professionals, management, IS academics, and IS users' clients. ISD is an intentional change process focusing on a given work activity, and ISD methodologies must meet several conditions to achieve their change mission in ISD process (Lyytinen 1987). According to Lyytinen, the choice of a development methodology should take into account its sensitivity to the cultural, social, political, and moral aspects of system design.

When we try to identify different aspects of ISD in Africa when compared to for example Europe or North America, we are not suggesting that the differences are caused by the people or culture in Africa. Rather, we regard that the present situation of Sub-Saharan Africa as a poorest and technologically least developed continent is mainly caused by the inheritance of insecurity and antidemocratic governance initiated by the slave trade and colonial occupation (Korpela 1994, 1996).

As Friedman (1989, see Chapter 2) predicted and Mathiassen (1998) described, ISD in the 1990s has been characterised by inter-organizational issues, for example by the introduction of global networks. Applications are integral parts of business strategies and enable collaboration across organizational and national boundaries. In Nigeria, this is becoming a reality for example for the oil industry and the banking sector, but in reality, global networks do not yet reach organizations in Nigeria.

On a more general level, the situation of IS practices in Nigeria seem to be a mix of features or constraints that were relevant in the industrialised countries in the 1960s, 1970s, and 1980s. For example, hardware is proportionally expensive and IT is mainly used to automate the manual processes, but on the other hand, with the emergence of PCs in 1980s, the focus is on user-friendliness and graphical interfaces. The key skills of developers in Nigeria are in analysis, design, and collaboration, which were emphasised in our cases. However, these are not yet emphasised in education. The software companies in Nigeria use CASE tools and the integration of systems were considered important, and eventually, big companies are evaluating information systems as a part of their business strategy. Software companies need people with the knowledge of the latest architectural development. Thus, the phases of hardware – software – user-relations constraints that Friedman (1989) identified in industrialised countries are not relevant in Nigeria. On the contrary, Nigeria seems to experience all these constraints at the same time to some extent.

There are some special requirements (or principles, c.f. Kensing et al. 1998) we consider important to take into consideration during the ISD process, but which are not sufficiently dealt with in ISD methodologies or curricula. Some of the requirements concern the information system as an artefact, some requirements concern the process of ISD. However, the requirements are very much interconnected, and cannot be separated.

The first issue is *sustainability* of the information system, which we have taken as an essential part of the successful ISD process, and which I have chosen as one of the key concepts of my thesis (c.f. Kensing et al. 1998). In order to fulfil the expectations of the information systems as a tool for development, these systems should be sustainable. Particularly in a less affluent country, the long-term viability of information systems is essential, but the required infrastructure and support activities are often lacking. The development process and methodology should consider the appropriateness of the technology to the application environment and the availability of the local technological capacity to sustain its beneficial use. The issue is that the new system would not be abandoned when the development project has ended. (Korpela et al. 2000b, Mursu et al. 2001)

I have discussed and defined the concept of sustainability in chapter 2 in more detail. The technical sustainability also includes appropriateness of technology, which is one of the major issues in technical introductions.

The second issue is *affordability* of the social information system and the new technology used within it. Information technology is still quite expensive an investment for companies in Africa, so it is important that the scarce resources are not wasted. The benefits must exceed the expenses, and the organization must be able to afford to cover the running costs, maintenance, and further development of the system in the long run. Thus the technology must be suitable for both users and developers, and the application must be adapted to the actual conditions. The existing information system must afford the changes caused by the new technology. For example, it can be risky to follow the latest technological trends, even if it can be tempting. (Mursu et al. 2001)

The third issue is the *ethics and socio-economic justification* of information technology. Information systems cannot just make life easier for immediate users, in particular in a less affluent country. Technology is never a solution *per se*. Even if information technology cannot directly influence high-priority issues like health, education or agriculture, it should have an indirect multiplier effect. For example, computers do not cure the diseased or feed the hungry, but they can make healthcare delivery more efficient, more accessible, and more focused. (Korpela et al. 2000b, Mursu et al. 2001)

So far we have argued that information systems should be sustainable, affordable, and they should have an ethical and socio-economic justification. In order to reach these requirements, the one very important requirement in the process of ISD is *user participation*. User participation is very much emphasised in industrialised countries as well (see for example Participatory Design in Chapter 2, and Kensing et al. 1998). It could be assumed that organizational barriers, economic hardship and the general insecurity of life creates obstacles for cooperation in developing countries, but the experiences by Korpela et al. (1998) in Nigeria indicate that participation and cooperation are not only possible in a deprived African country, but a must. They conclude that a computer-based system will not survive the harsh socio-economic conditions without the dedication of its users. Dedication begins by having users involved. Bringing users into the development process alleviates problems like computer phobia and fears and thus increases the user organization's technological capacity to sustain the system and reach a positive socio-economic impact. In addition, even if the clients we interviewed had not changed their work procedures dramatically by the new system, the user organization and the people within it need to sustain the changes caused by a new technology in the long run, and this is only possible by dedicated users. Korpela et al. (1998) also argue that the participation must be extended from the immediate users to their clients, at least in the healthcare sector. They suggest *community involvement* in ISD. However, we stress the importance of involving the *management* into the process (see also Bødker 1996, Kensing et al. 1998b), particularly in developing countries since the existence of a hierarchical administrative culture. (Korpela et al. 2000b, Mursu et al. 2001)

Kensing et al. (1998) suggest ethnographic techniques and intervention for developers to learn about the user organization, and to generate realistic visions of future use of technology. They argue that by going back and forth between observing users' work practice and producing a description or interpretations (in informal way), IT professionals and users are able to develop an understanding of the current practices that are relevant in design. We agree that bringing ethnography to the design process helps to promote user commitment to the process.

In order to manage ISD projects, which have been disposed to fail in terms of schedule or budget also in industrialised countries, project managers should take *risk management* as a serious part of the process. I have chosen risk management as one of the key concepts of my study, since I consider it a very important issue in project management, especially in the harsh contextual environment what Nigeria represents. Risk management is considered important also in the industrialised countries nowadays, and there is no need to repeat the mistakes that have been done there.

We can conclude that to be successful, the ISD process in a developing country like Nigeria requires ISD methods and techniques that are highly *practicable*, when applied by systems developers with limited education, and who work under high financial, infrastructural, administrative, and time constraints, and with scarce resources. Thus the methods should be straightforward, rapid, informal, and flexible. Methods should also be adjustable to differences in wider socio-political context. (Korpela et al. 2000b, Mursu et al. 2001)

One could say that these requirements are also important in industrialised countries. People still resist computers and there can be wrong investments and so on. I agree, and I point out that industrialised countries can learn from the experiences of developing countries as well. In summary, this theoretical analysis provides the direction where we have to aim when developing suitable ISD methodology for Nigerian systems developers in the INDEHELA-Methods project.

7.3 Risk management and sustainability analysis – a comprehensive framework

The practical contribution of this study is related to what is expected to be included in the Lecture Notes. The Lecture Notes will be a collection of methods and practices that we have found relevant for Nigerian practitioners during the INDEHELA-Methods project. The methods and practices are based on the empirical data and the theoretical analysis, which in turn is based on the empirical data and the literature in general. Besides to be provided to Nigerian practitioners, the Lecture Notes have been planned to be tested in the MINPHIS project in Nigeria (see Introduction) and in the systems analysis courses and final year projects of Computer Science students at the Obafemi Awolowo University. In short, the Lecture Notes is a portfolio of methods addressing the following issues: 1. project planning and management, 2. requirement analysis, 3. risk analysis, planning for sustainability, and ensuring a positive socio-economic impact, 4. grand design (designing the main functionality of an information system), 5. technical implementation, 6. organizational implementation (putting IT into use), and 7. long-term support (Soriyan et al. 1999).

As indicated above, one requirement for proper ISD methods and techniques is that they are highly practicable to be used by systems developers. Rapid, informal, and flexible methods are better than tedious, formalised, and rigid ones. In addition, the process of ISD should value principles like sustainability of the would-be information system, and affordability of the work practice to a new social and technical system. This means that the new system will change the existing work practice in any case, and that must be considered in the process. Also the ethical and socio-economic justification of the new system must be considered, so that the new technology would also benefit the last people in the client-chain, not just the immediate users. In that way the role of IT to promote socio-economic or human development can be considered. User participation is the first requirement if we want to ensure the above mentioned principles.

One can argue that for example the justification is the problem of the user organization, not software developers', but on the other hand, what if the user organization is the Mafia (c.f. Korpela et al. 1998), who wants help for blackmailing or drug pushing? We argue that it is important for IT professionals to consider the outcome of the people's work in every case. Neumann (1995) also asks how much computer professionals should consider the societal implications of their work. He had noticed that even if many scientific institutions[1] consider it important to have that kind of responsibility, the computing profession encourages computer scientists to be narrow technocrats. According to Neumann, most computer science curricula pays little or no attention to the 'social impacts'. We agree with him that

> *"the context in which computer systems exist – who pays for them, who participates in their design, how they are used, and how they are viewed by policymakers – is at least as significant an indicator of the systems' impact and value as are technical factors. Computer scientists are not the only ones who should consider the context of computer-system use, but their expertise imposes as special obligation to participate actively in the discussion"* (Neumann 1995, p. 278).

My part of the Lecture Notes should provide methods or tools for *risk management* and *sustainability analysis* in the ISD process. Thus my idea is to extend the traditional software project risk management ideas to also cover sustainable use of IT. Neumann (1995) offers a very comprehensive overview of risky situations caused by combinations of multiple hardware faults, system failures, and human mistakes. However, he concentrates mainly on complex hardware and software systems, even if he takes a human perspective into consideration, the use situation. Eventually, he does not provide any tools to manage these risks.

Even if we are talking about risk management, the issue is to make an ISD process successful. The truth is that people must be careful when developing, operating, and using computer-based systems (Neumann 1995), but when considering risks *beforehand*, we can concentrate on how to make a project successful.

---

[1] For example the executive director of Computing Research Association Rick Weingarter, the ACM Code of Professional Conduct, the director of National Science Foundation Walter Massey, and the National Academy of Science. (Neumann 1995)

TABLE 27 Risk categorization of risk factors in Nigeria, based on Cule et al. (2000, p. 68) model (bolded risks are within the top 19 risk factors)

INSIDE RISKS

Self	**Task**
• Not managing change properly • Lack of effective project management skills • Lack of effective project management methodology • Improper definition of roles and responsibilities. • Poor risk management. • **Choosing the wrong development strategy** • **Misunderstanding the requirements**. • Inadequate documentation of user requirements • **Lack of "people skills" in project leadership** • No planning or inadequate planning.	• Bad estimation. • **Lack of effective development process/methodology**. • **Trying new development method / technology during important project** • Lack of quality standards • Lack of computing literature • **Lack of required knowledge/skills in the project personnel** • Insufficient/inappropriate staffing. • **Lack of available skilled personnel**. • Introduction of new technology. • Inappropriate technology • Poor team relationships

OUTSIDE RISKS

Client	**Environment**
• **Lack of top management commitment to the project**. • Lack of client responsibility, ownership, and buy-in of the project and its delivered system(s). • **Failure to gain user commitment** • Customer's staff turnover • Failure to manage end user expectations • Lack of adequate user involvement • Lack of cooperation from users • Growing sophistication of users leads to higher expectations • Lack of appropriate experience of the user representatives • Customer's ability to react to change • **Inadequate user training** • Negligence of agreements • **Unclear/misunderstood scope/objectives**. • Scope creep • Lack of frozen requirements. • New and/or unfamiliar subject matter for both users and developers • **Under funding of development** • **Artificial deadlines** • Lack of experience of the user management	• Political climate in the country, including economic situation • **IT awareness in the country** • **Erratic and unreliable communication network** • **Energy supply** • Tertiary institutions • Poor copyright / intellectual property right protection • A climate of change in the business and organizational environment that creates instability in the project • **Changing scope / objective** • Unstable corporate environment. • No investments to IT. • **Huge capital requirements** • Poverty of software companies • Staffing volatility • **Importation of foreign packages**

However, we started from the model by Cule et al. (2000) for risk management, which has been introduced in Chapter 2, '*a risk categorization and behaviour model*'. Cule et al. categorized their risks in four groups, starting from the idea that some of the risks are under the project managers' control, when some are out of their control. We have categorised the risk factors identified in Nigeria accordingly (TABLE 27).

Some risks fall in the 'borderline' class. Somebody may feel that a particular risk may belong in another classification, and some risks may have a contextual element, but the purpose of the model is to help developing mitigation strategies relevant to the category rather than to each individual risk in isolation (Cule et al. 2000).

All in all, we found the risk categorisation based on Cule et al. (2000) model purposeful also in Nigeria. So there is no sense in trying to create new models for that.

The category of *Self* includes risks concerning a project manager's capabilities to manage software development projects. The responsibility for a successful process is demanding, thus *self assessment* cannot be ignored or underestimated. The identified risks in this category are quite critical, and project managers should ask themselves questions that are aimed at surfacing the risks in this category (Cule et al. 2000). The strategies to handle these risks can vary according to the situation, from assessment by upper management to benchmarking. The need for further education concerning management issues is essential to recognize.

Risks in the category of *Tasks* should be under the *control* of the project manager. The project manager should be able to organize and manage the activity inside the organization concerning the project task. He or she should be able to recognise the possible deficiencies in her team or means, and do the corrective or compensating actions accordingly. The project manager can apply for example work development methods like ActAd within this category.

The *Environment* risks are quite outside the project manager's control. The actions that the project manager can take here mainly include in keeping abreast of the environmental risks in order to maximise the possibilities for response, *monitor* the environment (Cule et al. 2000). In Nigeria the environmental risks are more visible and threatening when compared to the industrial countries. Cule's et al. idea of the monitoring responsibility requires that the project manager must be knowledgeable about what is going on in the country, and globally as well. Instead of just reading technical papers, the project manager should read the industry trade press and join conferences and seminars. In Nigeria, being active for example in the COAN (Computer Association of Nigeria) and work together with other software companies, they can try to influence government actions to improve the infrastructure and other investments. They can also work together to improve the reputation of local companies against foreign importation. In order to influence the environmental risks, the project manager should be active, politically and professionally. The environment risks also concern client organization, and these risks should also be considered in the client organization.

The risks involved in *Client* and its *relationships* are perhaps the most challenging, since these issues are hard to learn from technical books or schools. The 'client' includes people who fund the system and people who use the system, and they are essential to the successful outcome of the project (Cule et al. 2000), so it is critical in how to *relate* to these people. The responsibility of the project manager, together with top management in both site, is to create this relationship as a long-term relationship. Cule et al. discuss Relationship Management. We would also increase the 'clients of the client' to the 'client' group, since the impact of the new system concerns also them. Looking at the risk items in this category, the experiences in the case descriptions, and the requirements for a successful ISD process, the user relationships should be emphasised, starting at the very beginning of the project. Within this category, the issues like sustainability, affordability, ethical justification, and user participation can be included.

In this category, client relationships, we can extend the risk management model by Cule et al. (2000) to include also *sustainability analysis*. We apply the definition of sustainable technology by Oyomno (1996), presented in Chapter 2. We also apply the Activity Analysis and Development model and its basic elements, which were presented in Chapter 3. FIGURE 60 illustrates the extended model for risk management and sustainability analysis to be used in ISD projects.

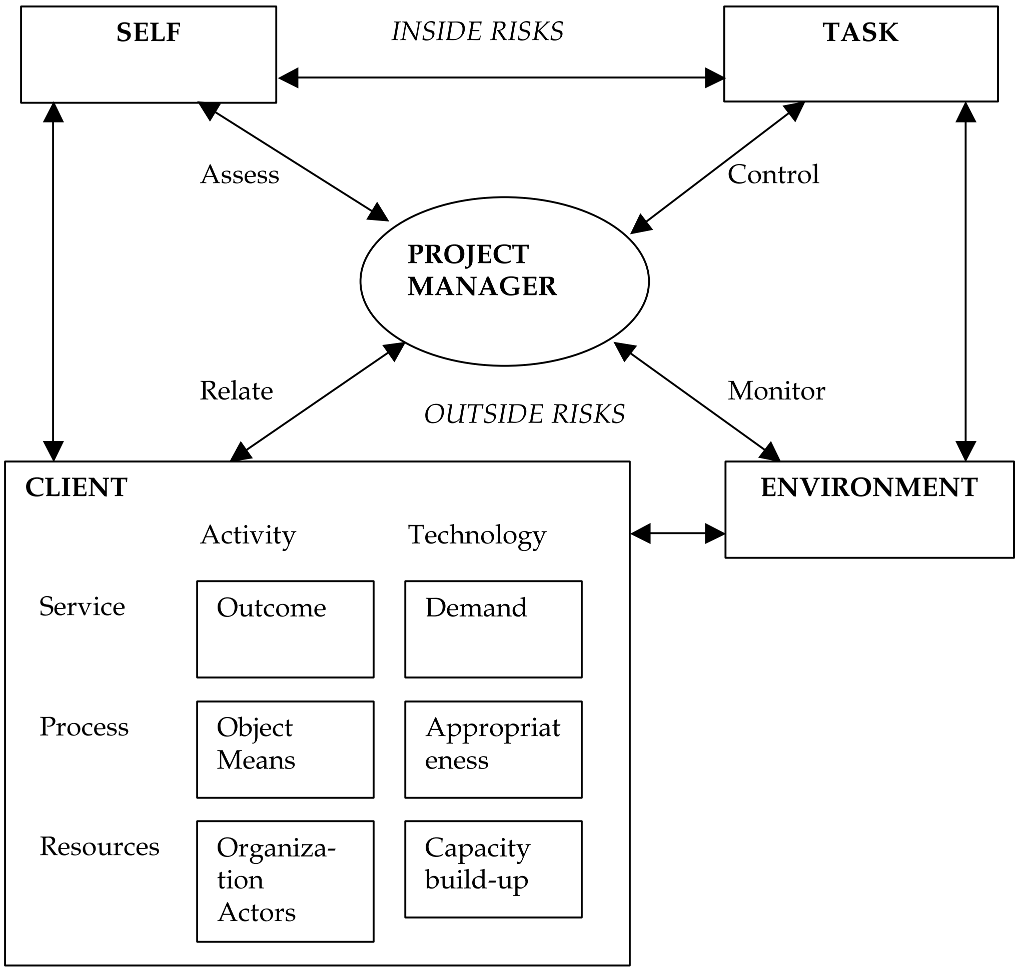


FIGURE 60 Risk management and sustainability analysis model

Sustainability analysis in the client organization is based on two levels of analysis. Firstly, the user organization is analysed on an activity level, where an object and an outcome is identified, means of work and means of coordination and communication, as well as actors in individual and group levels are described. The idea is to analyse the current activity, as well as the future activity. The activity analysis is related to technological analysis, which is the second level of analysis. The intended technology is evaluated in terms of outcome (what is the demand of technology for the service), object work system, and means of it (what is the appropriateness of technology for the process), and organizational structure and actors (do the organization have enough resources for the capacity build-up of new technology).

In the whole model, all the categories affect other categories. For example, in Appendix 5, we have presented the sustainability factors collected from IT experts in Nigeria, and they also emphasised the development process (task category) as a part of ensuring the sustainable use of IS. The other factors were divided according to Oyomno's model (demand, appropriateness, and capacity build-up). Thus there might be a need for a work development process within the software development process in the task category. Also the environment affects the client's organization as much as the task or the project manager herself. For example when developing work practises within the software company, environmental analysis (socio-economic) is an important level. Hence we have added two-edged arrows into the figure to illustrate the integration.

Sustainability analysis should be done in the very early phase of the project, as well as in the middle of the project once in a while, and at the end. It consists of steps, where certain factors in the user organization are checked, using ethnographical methods as much as possible. The steps can be applied according to the phase. User organization can include the analysis as part of the possible work development process (see Chapter 3). The steps have been originally adopted from the Primary Health Care Management Advancement Programme by Reynolds and Stinson (1993).

Step 1. Introduction: In this step the basic elements of the project are defined, like who are the user(s) and who are the clients and what is the purpose of the application. This is part of the project management activities anyway at the very beginning of the project (or before the project even begins formally).

Step 2. Sustainability objectives: In this step the sustainability objectives are defined, like what would the desired outcomes of the service be in the long-term, and what would be the intended impacts of new technology for the service or for the activity in general. These strategic objectives should be analysed in the early phase as well.

Step 3. Analyse factors: In this step the sustainability factors are analysed in terms of threats and opportunities, and financial implications. For each factor strategies for dealing with threats must be worked out. The factors are described in TABLE 28.

Each of the factors must be analysed in order to clarify the changes in the existing system caused by the new information technology. There should emerge a shared understanding of the existing system and vision of the new system and its benefits between client organization and developers.

TABLE 28 Sustainability factors of IS in use

	Activity	New technology
Service	**Outcome (present / new)**: What is the outcome of an activity? What is the service to be facilitated with new technology? What are the problems with the current system? How it relates to other services? How critical is this service or outcome? How would new technology change the outcome or service? Who would be the ultimate beneficiaries of the new system?	**Demand:** To what extent is technology put in order to proper functioning of organization? What is expected of productivity as a result? Value of output from the technology?
Process	**Object, means (present /new):** What is the process to be changed? What are the current means used in the process or activity? What means would be replaced by new technology? How would the process change by new means? What is the affect of change of means to other means in other services? Would the new means change the service chain somehow?	**Appropriateness:** How appropriate is the new technology to the organizational settings? What is the cost-effectiveness in terms of quality of information obtained, and extent to which jobs are enriched? What is the affordability in terms of financial and human resource requirements, and changes in organization? And what is the suitability of new technology in terms of operational simplicity, flexibility, maintainability and robustness?
Resources	**Organization, actors (present / new):** Who are the users of the existing system? Who would use the new technical system? Who would maintain the new system? Who would support hardware or software problems? How much is there already is needed knowledge for a new system? What other stakeholders would be involved in the new system? Would the new system change the structure of current org.?	**Capacity build-up**: How prepared is the user company to utilize effectively existing and new technology? How ready is the infrastructure for new technology? How ready are the entrepreneurial, technical, managerial, intellectual, institutional, socio-political, cultural, and physical resources for new technology?

Step 4. Strategic assessment: After the threats have been identified, as well as opportunities, for each factor in TABLE 28, proper responses must be considered. The opportunities must be analysed, and the threats must be neutralised, or at least mitigated. It is risk management in the user site.

Step 5. Strategy, action plan: All the responses are now collected together to form a strategy. Part of the strategy is focused on the new computerised system and in the developers' responsibility, part is focused on the activity itself, and thus the main responsibility is for the user organization. The strategy should be concreted to an action plan, with financial assessment.

The sustainability analysis can be combined to risk management practices. For example, what Boehm (1989) suggests as a simple *Risk Management Plan* can be connected (and modified) to the above steps: 1) Identify the risks that threatens the project's success at the moment (with a help of the model in FIGURE 60), 2) present a plan for resolving each risk item, 3) update the list of top risk items, plan, and results monthly, 4) highlight the risk-item status in monthly project reviews, compare with previous month's rankings, status, and 5) initiate appropriate corrective actions.

The essential issue of course is that the sustainability analysis must be done together with the client site. *User participation* is the precondition for the success of such analysis. The user organization is responsible for part of the actions resulted from the analysis. Together the risk management model and the sustainability analysis model would form a framework focusing on the total lifecycle risks. The whole framework should be embedded to the project management activities. The model is suggested here for the first time, thus it needs to be tested by the practitioners to evaluate its value. Further elaboration could be done according to the possible and desired comments of practitioners, students, and other researchers.

7.4 ActAd in IS research

The main methodological contribution is based on the use of the activity theoretical approach to IS research. In the case studies, the main unit of analysis was a work activity, and the idea was to study how information systems are developed in Nigeria in practice. The framework, called Activity Analysis and Development (ActAD), was used both as a research framework and for analysis. The background of the framework is explained in Chapter 3 and the way we used it in the study is illustrated in Chapter 5.

We can summarise that the methodological contribution had two main aspects. Firstly, the suitability of the activity theoretical approach and further, the ActAD framework, for IS research was tested and proved, and secondly, the ActAD framework was elaborated somewhat to make it more appropriate and usable for ISD research.

It has not been common to use work activity as the basic unit of analysis in IS research, but in order to study ISD methods and practices, the activity theoretical approach seemed to be usable. The ActAD framework provided a practical tool to design case interviews in that given context. However, based on the ActAD, the focus was on the collective work activity, rather than individual or group actions (c.f. Nurminen & Torvinen 1997 and the concept of role), purposely. Usually the application of the activity theory in the use and design of IT has focused on individual actions in human-computer interface design, or on analysing collective work on a rather general level (Bertelsen &

Bødker 2000). We argue that in order to truly operationalise the activity theory in the IT field, the approaches of 'designing software artefacts for individuals', viz. software engineering (SE), and 'developing work activities', meaning ISD, need to be linked together (c.f. Korpela et al. 2001a). I have discussed the difference between these two approaches in Chapter 2. We have suggested (Korpela et al. 2001a) that *use case modelling* would provide 'a bridge' to combine the activity level and action level of analysis, in the requirement phase. Thus we can argue that the activity theory is practicable in the activity level of analysis, but in order to operationalise the action level of the analysis needs some other tools, like use case modelling. However, going to the action level of the analysis was beyond the purpose of this thesis. After all, the question was not about work development, but to obtain knowledge about practices.

In addition to the action level, the other deficiency of the activity theory can be identified as being on the inter-activity level of the analysis and on the social level of the analysis. In Engeström's model the network of activities is analysed based on contradictions, but we emphasise the importance of *means of networking between* activities, in addition to the means of coordination and communication *within* activities. Also the *societal level* of the analysis, including the *politico-economic level* of the analysis, is necessary in order to understand the central activity in question.

The activity theory emphasises the importance of *the history* of an activity under study in analysis. We argue that the central activity, as well as the history of the activity, is impossible to understand if the socio-economic and political context of the given environment is ignored, also on a historical level. I admit that the historical analysis of the cases in this study could be better, but I also argue that even the short historical review of the country and the continent helps to see the reasons behind the problems we found in our cases. Even if my study focuses more on the social reasons instead of cultural reasons, for example the certain corruption issues are easier to understand if we knew the colonial history of the country. ActAD as a research framework does not provide actual tools to study social, political or economical context. The methods should be applied from other theories, and should also have been applied to this study in order to make the analysis more rigorous. Nevertheless, the extended level of the analysis in our study brought the relevant information into the picture.

When elaborating the ActAD method during the INDEHELA-Methods project, the main contributions are presented in Chapter 3, when expanding the framework. My own contributions relate to the means of networking and the improved checklists included in the structural analysis of central activity. In the case descriptions I have illustrated the means of networking between activities explicitly, not just in terms of contradictions or sources of means. It means the coordination and communication between the central activity and its nearby activities. The improvement of the checklists within the structural analysis was based on the first experiences in the interviews. Initially the checklist was divided into two parts, the composition of the central activity and the network of activities, as presented in Chapter 3. We combined the questions since there was a feeling of repetition during the interviews when discussing separately about the central activity and the network. The checklist we ended up with to use is presented in Chapter 5, where I have described how the case studies were conducted.

8 CONCLUSION

In this conclusion I briefly evaluate the obtained results by combining them to the initial research objectives and research questions. Do the results answer the stated questions? The reliability and validity of the results are also evaluated, and the limitations of the study. After that the suggestions for further research are presented, and finally, I will summarise the contributions of the study, and give a short summary of each chapter of this thesis.

8.1 Evaluation of the results

At this point of the study it is reasonable to evaluate the results in terms of reliability and validity, as well as the whole purpose of the study. In addition, the limitations of the study should be identified.

The *overall objective* of the study was to indicate, how risks in information systems development projects can be identified, monitored, and mitigated, and how sustainability of information systems can be facilitated by improved ISD methods. Based on the risk study and risk literature I ended up with recommending *a risk categorization and behavior model* by Cule et al. (2000) for risk management. It is more of a method to help assess and evaluate risks, rather than a strict tool for risk management, which explains how to manage each risk factor. Risks vary from one project to another, and no risk management method or strategy is suitable for every project. Thus project managers have to decide how to manage or mitigate risks in each case, and the model gives good directions with which to evaluate these risks. It is quite comprehensive, but still simple. I extended the model to identify a client more carefully as a user of the would-be information system. By this extension I aim to include sustainability analysis to the risk management task. The s*ustainability analysis model* is mainly based on literature, with some support from the empirical results.

The risk management model I recommend to use is based on long research by experienced researchers, and our results supported their categorization. I do not see any point to invent this wheel again, since their model seems reasonable for me. Even if the ranking phase of the Delphi method in our risk study did not reach a consensus, it does not mean that the results of the risk study are unreliable. The collection of 51 identified risk factors is based on 39 responses of experienced IT experts in Nigeria, so the number of responses is quite comparable with the results of Cule et al. (2000) in the US, Finland, and Hong Kong. Actually, the proportional number of respondents of all the IT experts in the country is quite big in our study, when compared to the other studies in the industrialized countries where the number of IT experts is much bigger in general. However, the ranking of the most important risk factors is only a qualified truth, but the collection of the 19 risk factors have aroused interest among other researchers in the field of informatics for development. I have considered the results of the differences of risk factors between Nigeria and the industrialized countries interesting and valuable knowledge. The limitations of the risk study are concentrated on the method used and its suitability in the research context.

The use of the Delphi method in the research context like Nigeria is a little complicated when compared to the industrialised countries. Summarising several reasons together we can conclude that the method – which is a good method as such – was not the best possible choice for that context, because the researchers were living in other towns or other countries, they had to rely on the students who were sent to Lagos to collect the responses. It was not possible to use posted questions or e-mails or phones, so the respondents were met face-to-face each time we wanted some response. This was a slow process since it was not possible for students to go there in the middle of semesters. In addition, going there by bus, moving from one place to another in Lagos, and to try to agree on the meetings took time, and people changed their work places. All together, in addition that the three phases took much more time that was appropriate, we lost most of our respondents. Eventually, we conducted the three phases, but the idea of the Delphi study and the consensus by respondents was lost. However, the results of the risk study as such are valuable.

The sustainability analysis model is based more on speculations on how sustainability issues should be considered during the ISD process. Oyomno's (1996) model gives a good aspect for technical evaluation; how appropriate the technology should be. However, as important it is to evaluate the changes in the work process caused by the new technology. For this kind of analysis I have applied elements from the activity theoretical approach. I did not manage to evaluate the sustainability factors well enough in the case studies. I should have applied some more theoretical model(s) for such an analysis. In the IS research such analysis is more focused on the computerized system itself (e.g. Neumann 1995), not on the work practices or work activity around the IT in a broad sense. The activity theoretical approaches provide methods for work development, so I tried to combine these two aspects: technological changes and work development within an organization. The model I have suggested needs to be evaluated, tested and commented by practitioners, as well as other researchers, for further elaboration.

The *overall research problem* was, whether there is a need for a new comprehensive perspective for sustainable development of ISs? I argue that yes, there is such a need. The success of the ISD process does not just mean a successful software project, but also sustained and long-term use of ISs. I argue that the model I have presented is the first step to such a comprehensive perspective.

It is also reasonable to evaluate how much, or how well, the results answered the *research questions* I have presented in the introduction. The research questions concerned about software development, information systems development, and information systems in use. These issues have not been studied much in developing countries before; how information systems are developed in practice. The problems concerning IS implementation and introduction have been identified in many articles (e.g. Waema 1996, Heeks 1999b). I have discussed these issues, firstly in the result chapter, but also in the discussion chapter. When we relate the case studies to the survey and the risk study, we can summarize that the results support quite well each other. Even if we evaluated our case companies to be a little bit more advanced than most other companies, the survey indicated that the average software company does not differ much for example from Gamma. And the differences did not affect much for the contributions we made.

In summary, I have evaluated that the case studies are quite representative and reliable in the activity level. The ActAD method provided a good practical framework to start to study the elements of ISD as a work practice, to concentrate on the relevant issues during the interviews.

There are still some limitations concerning the cases. Firstly, we were principally able to interview one person in the companies, who was on management level. Eventually we were able to discuss with one project manager and one consultant in Gamma, and one implementer in Beta. However, most of the descriptions of the companies are based on the responses of one person. It would have also been worth to discuss with some programmers or other employees on a lower level, or to have a group discussion after the first round of the analysis.

Someone might also miss an action or an operational level of analysis in the cases, which would have been possible by the discussions with other people than managers. However, as I have discussed earlier, it was not our purpose to go to that level.

Secondly, the analysis in the inter-organizational or societal and economical level was not rigorous, the analysis should have been done with some more theoretical methods. The same concerns the historical analysis, which is emphasized in the activity theory. I have discussed this earlier.

Thirdly, it must be noticed that the interviews were conducted by using a language that was not a mother language of any of the participants. The limited understanding of each other’s English, particularly the different accents of the Nigerian and Finnish partners, have surely affected the results. Not in the sense that there would be some mistakes, but I have surely missed some interesting details. I argue that it would have been totally different to do the interviews in

Finland for me, not because of the heat or the traffic or the infrastructure, but because of the language.

On the other hand, as a foreigner I was able to notice issues that would have been too obvious for the Nigerian researchers, and provide my western experience to bring different perspectives. This leads me to an issue that I do not consider as a limitation, even if I have received comments concerning the lack of it. It is the lack of cultural speculation of problems and behavior. Firstly, I do not have the education to evaluate the cultural issues, and I did not recognize many issues that can be considered cultural and have affected ISD as a work practice. Cultural differences play a subordinate role in explaining differences in systems development between Nigeria and Finland for instance (Korpela 1994). Secondly, if I had started to evaluate the problems in ISD culturally, they would have turned to *my problems*, not the Nigerian software companies' problems. When I worked in a software company in Finland, I did not consider the Finnish culture as a problem for me, but maybe I would have experienced cultural problems in a Nigerian software company. After all, we are trying to provide help for *the problems of the Nigerian software companies*. On the other hand, there are 'cultural' differences within one country or between different professionals. It can be a 'cultural shock' for a systems developer in Finland or in Nigeria to go to the hospital work environment to do some analysis. Thus cultural learning as such is an inevitable part of all systems development work (Korpela 1994).

What I have written above about the evaluation of problems is in contrast with the results of the sustainability factors and the model I have suggested. The model is based more on literature, because the amount of knowledge about the use of ISs in user organizations is limited to two interviews. I have discussed this already above. However, the client interviews were very interesting and gave additional value to the cases.

Since the main differences in all our results seem to be more in a socio-economic context than in software engineering or other technical issues, and the identified problems in our studies are very similar when compared to literature of IT in developing countries, we can presume that our results are quite applicable to other developing countries as well. In addition, concerning the special requirements identified in the discussion chapter, they can also be found important in industrialized countries, like sustainability, justification, and user participation. In that sense, the risk management and sustainability analysis model that I have suggested here can be applied in industrialized countries as well. Considering ISD also as a work development process in user organization is similar in developing countries as well as in industrialized countries.

8.2 Further research

It would be a waste to first conduct this kind of research that we have conducted in the INDEHELA-Methods project, and then just move to some other business. That is not the purpose and at the time of writing this thesis we are working hard to develop funding applications for further research. The work started with the thesis by Korpela (1994), where he said that his study was intended to be a preface for a long-term research programme. My work continued that research, and I really hope that the work will continue. As can be noticed from the publication years of our theses, the research area is not the easiest to raise funding for. However, we can identify some further research interests. Before that we have to finalize the Lecture Notes to be applied in the MINPHIS project and in the university education at OAU.

The most important part of further research would be to test in practice the risk management and sustainability analysis model I have suggested in this thesis. Firstly, the model could be tested and further elaborated in the MINPHIS project by using the action research method. Secondly, the model could be tested and elaborated by some Nigerian software companies, by using action research or case study methods. The important part would be to study the use of the sustainability analysis model in the client site, and to study the user participation in the ISD process.

The second very interesting research area concerns the ActAD method and its further application. It would be nice to conduct a work development process with some Nigerian software company. The adoption of our Made-in-Nigeria ISD methodology could be embedded in that process. In addition, to extend the model to include an action level, and socio-economic level of analysis, and to operationalise the model in that sense, is the next phase in that area.

The third research area that would interest me is to study the project management practices in Nigeria software companies in more detail. This is in relation to the risk management issues. It would be interesting to identify the strategies project managers and top managers are using when managing projects in such a constraint context.

The fourth very interesting research area would be to study global software service chains, where software development and use takes place in different socio-economic and cultural contexts. In this kind of chain the means of networking and management methods must take into account the differences in organizational, cultural, political, and socio-economic contexts. Besides practices and problems, it would be interesting to test how the risk management and sustainability analysis framework could be applied to cross-cultural settings.

The fifth research area would be to test the model in some Finnish or European software companies, rather with the process of work development within the company. The results and lessons that we have learnt from Africa could be transferred to an industrial context as well. As I noted in the introduction, Africa forms a mirror that lets us look at the ethics, constraints, benefits, and drawbacks of information technology from a global point of view.

8.3 Contributions and conclusion

The main concepts of this thesis include information systems development, risk management of software development, and sustainability analysis of information systems in use. Accordingly, the main objectives were to indicate, how risks in information systems development projects can be identified, monitored, and mitigated, and how sustainability of information systems can be facilitated by improved ISD methods. This thesis is part of the larger INDEHELA-Methods project, which aims at developing a Made-in-Nigeria methodology for improved ISD process. The purpose was to first provide empirical knowledge about ISD methods, techniques, and practices in Nigeria, and accordingly develop appropriate methods for improved ISD. This thesis focuses on risk management and sustainability analysis.

The study was intended to provide some knowledge contribution, as well as theoretical, practical, and methodological contributions, as identified in the introduction. The *knowledge contribution* concerns issues that have not been studied much in Africa before. There were no empirical studies of software industry in Africa, or information systems development practices in software companies in Africa. This study provides knowledge about ISD work practices, software development risks and problems in Nigerian software companies. Also the user organizations' perspective has been considered, especially in terms of sustainability of ISs. The INDEHELA-Methods project also provides a profile of the software industry in Nigeria in general. Thus this study provides additional information or value to the fields of information systems development, and especially information systems development in developing countries, in terms of software development practices, risks, and failures. In addition, the study clarifies how developed the Nigerian software companies are to provide information systems for development.

The *theoretical contribution* of this study mainly combines ISD theories with the activity theory, and extend the risk management theories to cover the use of ISs. In the light of IS research, the activity theoretical approach we applied tended to have some critical philosophical tendencies in its principles, like historical and contextual conditions (cf. Orlikowski & Baroudi 1991). However, in practice the study reminded more or less as interpretive in nature. For example the historical and contextual conditions were not studied rigorously enough. To my mind, the philosophy of pragmatism (Robey 1996, Wicks and Freeman 1998) is worth emphasising in an environment like Nigeria. Theoretical foundations for research and specific research methods should be justified by research aims, or purposes, not a dominant paradigm. Theories and methods are justified on pragmatic grounds as appropriate tools for accomplishing research aims. Also ethics should be considered in the research efforts.

The *practical contribution* includes methods for risk management and sustainability analysis, intended to be provided to the Nigerian software companies in the first place. However, the developed framework is not rejected to an African context, but it can be transferred to other developing or industrial countries as well. The idea of the framework is that risk management and sustainability analysis should be an integral part of a project management activity, and that it must be based on user participation. ISD is considered as work development activity in user organization at the same time.

The *methodological contribution* of this study is mainly the way we have applied the activity theoretical approach to IS research. The research framework we used – ActAD – has been further elaborated and tested. The means of coordination and communication between the activities – in the network of activities – are taken as an important part of the activity analysis, thus providing a new aspect to the activity theoretical approaches on theoretical level. In addition, in order to get the ActAD method more comprehensive and rigorous there is a need for some applied methods from other theories, for example for societal level of analysis. In this thesis there is no need to start to speculate which methods, I will leave that to further studies.

As a conclusion, this study consisted of a theoretical part including pertinent literature reviews, an empirical part including the results of the study, and finally the contributions of the study. Chapter 1 introduced the research objectives, research questions, and the overall research design, including intended contributions. The introduction chapter also explains the motivation of the study, and the theoretical background. The theoretical background can be divided into three parts: information systems and information systems development, activity theory, and information technology in developing countries. Thus the theoretical background includes the research interest, research approach, and research context.

Chapter 2 consists of literature reviews concerning the relevant issues in information systems research in relation to the research questions. Firstly, the concept of an information system is presented in terms of research and practice. I consider that part important because we define information systems as social systems, rather than purely technical systems, and the review of the development of IS research and practice sheds light on the situation in Nigeria, and also how we have approached the issue. Our research approach is qualitative, even if it includes some minor quantitative features in both surveys. The related research on ISD is also examined. Secondly, since my research interest initially was in software project risks, the success or failure of the ISD projects was a logical movement in that direction. In developing countries the success of ISD project has a broader meaning, since IT has been expected to promote socio-economic and human development. The sustainable use of information systems has become an important part of the definition of successful IT introduction. In that sense it was reasonable to combine the two sides of the coin; risks in software development and risks in the use of information systems.

Chapter 3 provides an overview of the activity theory and how it has been applied to the information systems research before. The Activity Analysis and Development (ActAD) method is presented at the end of the chapter, since the method has been used as a research framework and for analyzing the cases in this study. Chapter 4 illustrates the research context in more detail. Firstly, it is reasonable to understand what we mean by development, if we want to see IT as an important part of the development. The history of the continent and the history of Nigeria is important to be presented in that sense, even in short, in order to understand the present situation and the future. Information technology in Nigeria is illustrated for example by presenting the results of the survey conducted within INDEHELA-Methods project. Chapter 5 presents the methodological foundations and describes how the study was conducted in practice.

In chapter 6 the results of my part of the INDEHELA-Methods project are presented. Firstly, the results of the risk study are presented phase by phase. The collection of 51 risk factors in Nigeria complete the known risk factors in the industrialized countries. The ranking of risk factors did not reach a consensus in Nigeria, but however, we could compare the results to the studies conducted earlier in the US, Finland, and Hong Kong. The results indicated that the differences in software development risk factors are mainly in a socio-economic context. In addition, we could use the results to suggest a method for risk management. Secondly, the case studies of software companies are presented, with addition of a couple of client interviews. The case descriptions illustrate quite well the ISD practices in the Nigerian software companies. We compare the cases with the survey of a 'typical Nigerian software company', and with the risk study. We can conclude that all together, we have obtained a pretty good picture of ISD work practices in the software companies in Nigeria, and the problems they face.

In chapter 7, I have discussed the results of my study in the light of intended contributions. Firstly, I discuss the obtained knowledge and how the knowledge answers the research questions identified in the beginning. We can summarize that the differences in ISD between developing countries and industrial countries are based on a socio-economic context, and software engineering is more the same when compared to industrial countries. The software companies in Nigeria are capable of providing information systems for local organization. Only the resources for IT investments in Nigerian organizations in general are scarce. However, the majority of software companies provide services for the domestic market. Secondly, I have identified what we have found to be important requirements for appropriate ISD methodology in Africa. As we see IS more like a social system, we emphasis such issues as sustainability and affordability of IS, the ethics and socio-economic justification of IT, user participation, and risk management. In addition, ISD methods should be highly practicable. Thirdly, I present the risk management model applied from Cule et al. (2000), which has been extended by the sustainability analysis model. The whole framework of risk management and sustainability analysis must be tested next and evaluated by practitioners and other researchers, and finally, I discuss and evaluate the activity theoretical approach we have used in this study, the ActAD method. In order to develop the method more comprehensive and operational, there is a need to apply additional theories, especially on the action level and socio-economic level.

Finally, in this chapter, the results are evaluated in terms of reliability and validity, and the limitations of the study are noted. Further research interests are identified before the contributions and conclusion.

As a final conclusion I can admit that I was a little surprised to notice how professional and 'modern' the activities in the Nigerian software companies are. The truth is that software companies in Nigeria want, and are capable of providing software solutions for their clients. The infrastructure is still the biggest constraint to effective work, and the university education needs to improve much more. We agree that strategies for harnessing IT for development should include some actions by the government, but also the project management in the software companies has an extremely critical role for a successful ISD process, not just in terms of technological solutions or quick profit, but to promote development in the country. IT industry in Nigeria is young, thus it is extra challenging for software companies to take into consideration the *requirements* for a successful ISD process; information systems must be appropriate and sustainable, IT is not a solution as such, but it must have ethical and socio-economic justification, and its development must be based on user participation.

YHTEENVETO (FINNISH SUMMARY)

Tämä työ käsittelee tietojärjestelmien kehittämistä kehitysmaissa, tarkemmin Nigerialaisissa ohjelmistoyrityksissä. Lähtökohtana on ajatus, että ainakin organisaation toiminnan kannalta keskeisten tietojärjestelmien kohdalla on eduksi jos ne voidaan kehittää paikanpäällä siinä kulttuuriympäristössä, missä niitä tullaan käyttämään. Joka tapauksessa järjestelmän paikallinen räätälöinti käyttöönoton yhteydessä on usein tarpeen, varsinkin silloin kun sovellus on kehitetty muualla maailmassa, jotta järjestelmän soveltuvuus ja elinkelpoisuus olisi mahdollisimman hyvä. Esimerkit ovat osoittaneet että tietojärjestelmien soveltumattomuus muun muassa Afrikassa on johtanut näiden sovellutusten vajaaseen käyttöön tai väärinkäyttöön. Joskus sovellukset on jopa hylätty, koska ne eivät ole palvelleet käyttötarkoitusta, tai niitä ei ole osattu käyttää.

Modernin tietotekniikan on odotettu olevan se ase, jonka avulla kehitysmaiden ja teollisuusmaiden välinen hyvinvoinnin kuilu voidaan kuroa umpeen. Kuitenkaan tietotekniikan avulla muun muassa Afrikassa ei ole saavutettu juurikaan niitä kehityksellisiä tavoitteita, joita sille on asetettu. Monet ovat jo epäilleet tietotekniikan soveltuvuutta taloudellisen ja inhimillisen kehityksen edistäjänä kehitysmaissa. Mutta on muistettava, että tietojärjestelmä on organisaation tietojenkäsittelyn järjestelmä – siis sosiaalinen järjestelmä, joka pitää sisällään organisaation ja toiminnan ihmisineen ja jossa hyödynnetään paperipohjaista ja tietokonepohjaista tietotekniikkaa. Aina kun kehitetään uutta tietotekniikkaa tulisi ottaa huomioon se työympäristö mihin sitä viedään ja tarkastella asiaa ennen kaikkea toiminnan kehittämisen kannalta. Tietojärjestelmien kehittäminen on tietotekniikan kehittämisen lisäksi myös työn ja toiminnan kehittämistä. Siksi tietotekniikan pelkkä dumppaaminen muun muassa kehitysmaihin on monesti tuomittu epäonnistumaan, jos käyttöympäristö on jäänyt vaille huomiota. Järjestelmien paikallinen kehittäminen on siis merkittävää tietojärjestelmien onnistuneen käyttöönoton kannalta. Mutta aikaisempaa tutkimusta siitä, miten esimerkiksi Nigeriassa ohjelmistotalot toimivat, miten kehittyneitä ne ovat, ja onko niitä ylipäätään, ei ole tehty.

Tämän väitöskirjan tarkoitus on selvittää miten tietojärjestelmiä Afrikassa kehitetään, mitä toimintatapoja, työkaluja ja menetelmiä paikalliset tietojärjestelmiä kehittävät yritykset käyttävät ja soveltavat, ja minkälaisia ongelmia ne kohtaavat. Tietojärjestelmien kehittämisen lisäksi keskeisiä teemoja tässä väitöskirjatyössä ovat riskien hallinta sekä teknisten ratkaisujen elinkelpoisuuden analysointi jo tietojärjestelmäprojektin aikana, ottaen huomioon kestävän kehityksen periaatteet.

Tutkimusmenetelminä tässä työssä on sovellettu kyselytutkimusta, Delphi-menetelmää, sekä tapaustutkimusta. Kaikissa menetelmissä on käytetty metodina haastattelua, koska tietoliikenneongelmien ja epäluotettavan postipalvelun takia ei voitu turvautua tehokkaampiin metodeihin.

Tutkimuksen empiirinen aineisto koottiin neljän vuoden aikana (1998-2001) Nigerialaisilta tietojärjestelmäyrityksiltä, sekä heidän asiakkailtaan. Aineisto koostuu kolmesta osasta perustuen edellä mainittuhin tutkimusmenetelmiin. 1) Ensin tutkittiin mitä ovat tietojärjestelmäprojektiin liittyvät riskit, käyttäen Delphi menetelmää. Tutkimus oli toistotutkimus, jossa käytettiin samaa tutkimusasetelmaa kuin oli käytetty aikaisemmin tehdyssä riskitutkimuksessa Yhdysvalloissa, Suomessa ja Hongkongissa. Delphi-tutkimukseen osallistui yhteensä 11 yritystä. 2) Tapaustutkimusta soveltaen selvitettiin minkälaisia tietojärjestelmien kehittämisen menetelmiä ja työtapoja Nigerialaisissa yrityksissä käytetään, ja minkälaisia ongelmia yritykset kohtaavat. Tutkimuksessa haastateltiin kolmea ohjelmistoyritystä, sekä kahta heidän asiakasyritystään. Tapaustutkimuksessa käytettiin taustateoriana ja tiedonkeruumenetelmänä toiminnan teoriaa, tarkemmin 'Activity Analysis and Development' (ActAd) menetelmää. Työn aikana menetelmän käytettävyyttä ja soveltuvuutta tietojärjestelmätieteen tutkimuksessa testattiin ja edelleen kehitettiin. 3) Lisäksi selvitettiin kyselytutkimusta käyttäen ohjelmistoteollisuuden kuva Nigeriassa. Kyselytutkimukseen osallistui yhteensä 103 yritystä. Tähän tutkimukseen väitöskirjassa viitataan projektin julkaisujen kautta, ei varsinaisina tuloksina tähän väitöskirjatyöhön liittyen.

Nigeriassa on noin tuhat tietotekniikkayritystä, ja kyselytutkimuksen perusteella voimme arvioida että niistä noin 150-200 tarjoaa ohjelmistokehityspalveluja tai tietojärjestelmäpalveluja, tietokonemyynnin ja koulutuspalvelujen lisäksi. Tulosten mukaan ohjelmistoyritykset ovat ammattitaitoisia ja käyttävät ajanmukaisia menetelmiä ja työkaluja. Ohjelmistotuotanto on siis hyvin samanlaista verrattuna teollisuusmaiden ohjelmistotuotantoon. Suurimmat eroavaisuudet ja ongelmat löytyvätkin siitä ympäristöstä ja sosiaalisesta kontekstista, jossa tietojärjestelmien kehitystyötä tehdään, kun tarkastellaan yhteistyötä käyttäjäorganisaatioiden kanssa. Ongelma-alueita ovat muun muassa maan heikko infrastruktuuri, yliopistokoulutuksen puutteellisuudet, käyttäjien kokemattomuus ja tietämättömyys tietotekniikan käytöstä, sekä päättäjien kykenemättömyys nähdä tietotekniikan mahdollisuuksia ja luoda edellytyksiä ohjelmistotuotannolle, tai edes pitkäjänteisille tietotekniikkainvestoinneille. Ohjelmistoyrityksillä onkin haasteellinen tehtävä selvitä ympäristössä, joka ei kovinkaan paljoa tue heidän toimintaansa. Tämä asettaa projektien johdolle kovia vaatimuksia, koska teknisten ratkaisujen lisäksi myös ympäristön luomat riskit tulisi hallita.

Empiiristen tulosten sekä kirjallisuuden pohjalta tässä väitöskirjatyössä esitetään tiettyjä vaatimuksia tai periaatteita, joita pitäisi ottaa huomioon tietojärjestelmäprojekteissa, käytettävissä menetelmissä, sekä myös tietojärjestelmäkoulutuksessa. Näiden työssä esitettyjen periaatteiden mukaan tietojärjestelmistä pitäisi tulla soveltuvampia ja kehityksellisesti kestävämpiä käyttöympäristöönsä nähden. Lisäksi tässä työssä ehdotetaan mallia riskitekijöiden analysoimiseksi, sekä niiden tekijöiden analysoimiseksi, jotka vaikuttavat kestäviin tietoteknisiin ratkaisuihin, jo tietojärjestelmäprojektin aikana. Mallin soveltuvuutta tietojärjestelmäprojekteissa käytettäväksi menetelmäksi Afrikkalaisissa ohjelmistotaloissa ei ole testattu tämän työn yhteydessä, vaan se kuuluu osaksi tulevaa tutkimusta.

Väitöskirja perustuu INDEHELA-Methods tutkimusprojektiin, joka on yhteistyöprojekti Kuopion yliopiston ja Nigerialaisen Obafemi Awolowo University'n (OAU) kanssa. Projekti sai rahoitusta Suomen Akatemialta neljän vuoden ajan, vuosina 1998-2001.

REFERENCES

Achterberg J., van Es G., Heng M. (1991). Information Systems Research in the Postmodern Period. In Nissen, Klein, Hirschheim (eds.). *Information Systems Research: Contemporary Approaches & Emergent Traditions*. North-Holland: Elsevier Science Publishers. 281-294.

Akinde A.D. (1998). Speech delivered at the 2nd Made in Nigeria Software Exhibition in Calabar, 31st March.

Akinlade T.O. (1989). Information Technology in Nigeria: Problems and Prospects. *Information Technology for Development*. 4(3). 667-672.

Alabi G.A. (1994). *Case Study Effectiveness of Informatics Policy Instruments in Africa: Nigeria*. IDRC.

Alavi H., Shanin T. (eds.) (1982). *Introduction to the Sociology of 'Developing Societies'* . Macmillan.

Alter S. (1975). *A Study of Computer Aided Decision Making in Organizations*. PhD dissertation. Massachusetts Institute of Technology.

Alter S. (1999). A General, Yet Useful Theory of Information Systems, Communications of the AIS. 1(13). March. (15.2.2002) http://cais.isworld.org./articles/1-13/.

Alter S. (2001a). Are the Fundamental Concepts of Information Systems Mostly about Work Systems?. Communications of the AIS. 5(11). April. (15.2.2002) http://cais.isworld.org./articles/5-11/.

Alter S. (2001b). Which Life Cycle … Work System, Information System, or Software?, Communications of the AIS. 7(17). October. (15.4.2002) http://cais.isworld.org./articles/7-17/

Alter S., Ginzberg M. (1978). Managing uncertainty in MIS implementation. *Sloan Management Review* 20(1). 23-31.

Amin S. (1997). *Capitalism in the age of globalization*. London: Zed Books.

Anakwe U., Anandarajan M., Igbaria M. (1998). Information Technology Usage Dynamics in Nigeria: An Empirical Study. *Journal of Global Information Management*. 7(2). 13-21.

Andah B.W. (1992). *Nigeria's Indigenous Technology.* Nigeria, Ibadan: Ibadan University Press.

Andersen N.E., Kensing F., Lassen M., Lundin J., Mathiassen L., Munk-Madsen A., Sørgaard P. (1990). *Professional Systems Development. Experiences, Ideas, and Action.* Englewood Cliffs, New Jersey: Prentice-Hall.

Andersen P. (1991). A Semiotic Approach to Construction and Assessment of Computer Systems. In Nissen, Klein, Hirschheim (eds.). *Information Systems Research: Contemporary Approaches & Emergent Traditions.* North-Holland: Elsevier Science Publishers. 465-514.

Applegate L.M., McFarlan F.W., McKenney J.L. (1996). *Corporate Information Systems Management. Text and Cases*. Fourth Edition, Chicago: Irwin.

Asaro P.M. (2000). Transforming society by transforming technology: the science and politics of participatory design. *Accounting, Management and Technology* 10(4). 257-290.

Aulin-Ahmavaara A.Y. (1977). A general theory of acts, with application to the distinction between rational and irrational 'social congnition'. *Zeitschrift, allgemeine Wissenschaftstheorie* 2. Vol. VIII. 195-220.

Avgerou C. (1990). Computer-based information systems and modernization of public administration in developing countries. In Bhatnagar, Bjørn-Andersen (eds.). *Information Technology in Developing Countries.* North-Holland: Elsevier Scinece Publishers. 243-250.

Avgerou C. (1996). Transferability of information technology and organisational practices. In Odera-Straub M. (ed.). *Global Information Technology and Socio-Economic Development*. Nashua: Ivy League. 106-115.

Avgerou C., Land F. (1992). Examining the appropriateness of information technology. In Bhatnagar, Odedra (eds.). *Social Implications of Computers in Developing Countries*. New Delhi: Tata McGraw-Hill. 26-41.

Avison D. (1997). The 'Discipline' of Information Systems: Teaching, Research, and Practice. In Mingers, Stowell (eds.). *Information Systems: An Emerging Discipline?.* Berkshire: McGraw-Hill Publishing Company. 113-136.

Avison D.E., Fitzgerald G. (1988). Information Systems Development. Current Themes and Future Directions. *Information and Software Technology* 30(8). 458-466.

Avison D., Fitzgerald G. (1995). *Information Systems Development: Methodologies, Techniques and Tools*. 2nd edition. Berkshire: McGraw-Hill.

Avison D., Lau F., Myers M., Nielsen P.A. (1999). Action research. *Communications of the ACM* 42(1). 94–97.

Awad M., Kuusela J., Ziegler J. (1996). *Object-Oriented Technology for Real-Time Systems, A practical approach using OMT and Fusion*. Upper Saddle River, NJ: Prentice-Hall.

Bada A.O. (2000). 'Actually existing globalisation': A case study of IT and organizational change in a Nigerian bank. In the proceedings of IFIP WG 9.4 conference. *Information Flows, Local Improvisations and Work Practices*. South Africa.

Bannon L.J., Bødker S. (1991). Beyond the Interface: Encountering Artifacts in Use. In Carrol, John (eds.). *Designing Interaction*. Cambridge: Cambridge University Press. 227-253.

Bannon L.J., Kuutti K. (1993). Searching for Unity among Diversity: Exploring the 'Interface' Concept. In Ashlund, Stacey et al. (eds.). *INTERCHI'93: Conference on Human Factors in Computing Systems INTERACT'93 and CHI'93 Bridges Between Worlds*. New York: ACM Press. 263-268.

Banville C., Landry M. (1989). Can the field of MIS be disciplined? *Communication of ACM* 32(1). 48-61.

Barki H., Rivard S., Talbot J. (1993). Toward an Assessment of Software Development Risk. *Journal of Management Information System* 10(2). 203-225.

Baruah N. (2000). Computerizing tax administration in India – a case study of short-term wins. In the Proceedings of IFIP WG 9.4 conference. *Information Flows, Local Improvisations and Work Practices*. South Africa.

Baskerville R. (1991). Risk analysis: an interpretive feasibility tool in justifying information systems security. *European Journal of Information Systems* 1(2).

121-130.
Baskerville R., Stage J. (2000). Discourses on the Interaction of Information Systems, Organizations, and Society: Reformation and Transformation. In Baskerville, Stage, DeGross (eds.). *Organizational and Social Perspectives on Information Technology*. Boston: Kluwer Academic Publishers.1-12.
Baskerville R., Stage J. (2001). Accommodation Emergent Work Practices: Ethnographic Choice of Method Fragments. In Russo, Fitzgerald, DeGross (eds.). *Realigning Research and Practice in Information Systems Development, The Social and Organizational Perspective*. IFIP WG 98.2 Working Conference. Boston: Kluwer Academic.11-28.
Bell S., Wood-Harper T. (1990). Information Systems Development for Developing Countries. In Bhatnagar, Bjørn-Andersen (eds.). *Information Technology in Developing Countries*. North-Holland: Elsevier Scinece Publishers. 23-39.
Benbasat I., Goldstein D., Mead M. (1987). The Case Research Strategy in Studies of Information Systems. *MIS Quarterly*. 11(3). September. 369-386.
Benbasat I., Weber R. (1996). Research Commentary: Rethinking 'Diversity' in Information Systems Research. *Information Systems Research* 7(4). 389-399.
Bertelsen O.W. (2000). Design Artefacts: Towards a Design-oriented Epistemology. *Scandinavian Journal of Information Systems* 12. 15-27.
Bertelsen O.W., Bødker S. (2000). Introduction: Information Technology in Human Activity. *Scandinavian Journal of Information Systems* 12. 3-14.
Beynon-Davies P. (1995). Information systems 'failure': the case of the London Ambulance Service's Computer Aided Despatch project. *European Journal of Information Systems* 4(3). 171-184.
Bhatnagar S.C. (1992). *Information Technology Manpower: Key Issues for Developing Countries*. New Delhi: Tata McGraw-Hill.
Bhatnagar S.C. (1994). Strategies for exploiting the global software market: Indian experience. In Duncan, Krueger (eds.). *Linkage and Developing Countries, Information Processing '94*, Vol III, North-Holland: Elsevier Science. 38-45.
Bijker W.E., Hughes T.P., Pinch T.J. (eds.) (1987). *The Social Construction of Technological Systems*. Cambridge, Massachusetts: MIT Press.
Bilinski A.E. (1986). Systems Failure Models as a Result of Design Inadequacy. In Wise, Debons (eds.). *Information Systems: Failure Analysis*. Berlin: Springer-Verlag. 21-27.
Bishop S. (2000). Computer applications development at Barbados utility company. In the Proceedings of IFIP WG 9.4 conference, *Information Flows, Local Improvisations and Work Practices*. South Africa.
Bjerknes G., Bratteteig T (1987). Florence in Wonderland. Systems Development with Nurses. In Bjerknes, Ehn, Kyng (eds.). *Computers and Democracy*. Aldershot: Avebury. 279-295.
Bjerknes G., Bratteteig T. (1995). User Participation and Democracy: A Discussion of Scandinavian Research on Systems Development. *Scandinavian Journal of Information Systems*, 7(1). 73-97.
Bjørn-Andersen N. (1985). Conference Review: IS Research – A Doubtfull Science. In Mumford, Hirschheim, Fitzgerald, Wood-Harper (eds.). *Research*

Methods in Information Systems. North-Holland: Elsevier Science Publishers. 273-280.

Bjørn-Andersen N., Hedberg B., Mercer D., Mumford E., SolÈ A. (eds.) (1979). *The Impact of Systems Change in Organizations*. Netherlands: Sijthoff & Noordhoff.

Boahen A.A. (1989). *African Perspectives on Colonialism*. Baltimore: The Johns Hopkins University Press.

Boehm B.W. (1989). *Software Risk Management Tutorial*. Washington D.C: IEEE Computer Society Press.

Boehm B.W. (1991). Software Risk Management: Principles and Practices. *IEEE Software* 8(1), 32-41.

Boehm B.W., Ross R. (1989). Theory-W Software Project Management: Principles and Examples. *IEEE Transactions on Software Engineering* 15(7). 902-916.

De Boer S.J., Walbeek M.M. (1999). Information technology in developing countries: a study to guide policy formulation. *International Journal of Information Management* 19. 207-218.

Boland R. Jr (1985). Phenomenology: A Preferred Approach to Research on Information Systems. In Mumford, Hirschheim, Fitzgerald, Wood-Harper (eds.). *Research Methods in Information Systems*. North-Holland: Elsevier Science Publishers. 193-202.

Boland R. Jr (1991). Information Systems Use as a Hermeneutic Process. In Nissen, Klein, Hirschheim (eds.). *Information Systems Research: Contemporary Approaches & Emergent Traditions*. North-Holland: Amsterdam. 439-458.

Brooks F.P. Jr (1987). No Silver Bullet: Essence and Accidents of Software Engineering. *IEEE Computer* 20(4). 10-19.

Brooks S. (1974). *A Mythical Man-Month: Essays of Software Engineering*. Addison-Wesley London: Prentice-Hall.

Brundtland Commission (World Commission on Environment and Development) (1987). *Our Common Future*. Oxford: Oxford University Press.

Bubenko J. Jr., Langefors B., Solvberg A. (eds.) (1971). *Computer-Aided Information Systems Analysis and Design*. Studentlitteratur. Lund.

Buckingham R.A., Hirschheim R.A., Land F.F., Tully C.J. (1987). *Information Systems Education: Recommendations and Implementation*. Cambridge: Cambridge University Press.

Burrel G., Morgan G. (1979). *Sociological Paradigms and Organizational Analysis*. London: Heinemann.

Bødker S. (1987/1991a). *Through the Interface – a Human Activity Approach to User Interface Design*. Hilsdale, NJ: Lawrence Erlbaum Associates. Originate: Aarhus University: Daimi PB. 1987.

Bødker S. (1991b). Activity Theory as a Challenge to Systems Design. In Nissen, Klein, Hirschheim (eds.). *Information Systems Research: Contemporary Approaches & Emergent Traditions*. North-Holland: Amsterdam. 551-564.

Bødker S. (1996). Creating conditions for participation: Conflicts and resources in systems design. *Human Computer Interaction*, 11(3). 215-236.

Bødker S. (1997). Computers in Mediated Human Activity. *Mind, Culture, and*

Activity, 4(3). 149-158.

Bødker S., Ehn P., Kammersgaard J., Kyng M., Sundblad Y. (1987). A Utopian Experience. On Design of Powerful Comuter-based Tools for Graphical Workers. In Bjerknes, Ehn, Kyng (eds.). *Computers and Democracy*. Aldershot: Avebury.

Bærentsen K.B. (2000). Intuitive User Interface. *Scandinavian Journal of Information Systems* 12. 29-60.

Callon M. (1986). Some Elements of a Sociology of Translation: Domestication of the Scallops and the Fishermen. In: Law (ed.). *Power, Action and Belief: An New Sociology of Knowledge?*. London: Routledge & Kegan Paul.

Calloway L., Ariav G. (1991). Developing and Using a Qualitative Methodology to Study Relationships among Designers and Tools. In Nissen, Klein, Hirschheim (eds.). *Information Systems Research: Contemporary Approaches & Emergent Traditions*. North-Holland: Elsevier Science Publishers. 175-193.

Castells M. (1996). *The Rise of the Network Society*. Massachusetts: Blackwell Publishers Ltd.

Castells M. (1998). *End of Millennium*. Massachusetts: Blackwell publishers Ltd.

Charette R.N. (1989). *Software Engineering Risk Analysis and Management*. New York: Intertext.

Charette R.N., MacG. Adams K., White M.B. (1997). Managing Risk in Software Maintenance. *IEEE Software*. May/June. 43-50.

Checkland P. (1981). *Systems Thinking, Systems Practice*. Chichester: John Wiley.

Checkland P., Scholes J. (1990). *Soft Systems Methodology in Action*. Chichester: John Wiley.

Christiansen E. (1996). Tamed by a Rose: Computers as Tools in Human Activity. In Nardi (ed.). *Context and Consciousness: Activity Theory and Human-Computer Interaction*. Cambridge MA: MIT Press. 175-198.

Chua W.F. (1986). Radical Developments in Accounting Thought. *The Accounting Review* 61(4). 601-632.

Conrow E.H., Shishido P.S. (1997). Implementing Risk Management on Software Intensive Projects. *IEEE Software*. May/June. 83-89.

Corbridge S. (1998). 'Beneath the pavement only soil': The poverty of post-development. *The Journal of Development Studies*. 34(6). 138-148.

Cule P., Schmidt R., Lyytinen K., Keil M. (2000). Strategies for Heading Off IS Project Failure. *Information Systems Management*. 17(2). 65-73.

Curtis B., Krasner H., Iscoe N. (1988). A Field Study of the Software Design Process for Large Systems. *Communications of the ACM*. 31(11). 1268-1287.

Dahl O.J., Nygaard K. (1966). SIMULA – An Algol-based simulation language. *Communications of the ACM*. 9(9). 671-678.

Davis G.B. (2000). Information Systems Conceptual Foundations: Looking Backward and Forward. In Baskerville, Stage, DeGross (eds.). *Organizational and Social Perspectives on Information Technology*. Boston: Kluwer Academic Publishers. 61-82.

Davison B. (1983). *Modern Africa*. London: Longman Group Ltd.

Dhillon G., Backhouse J. (1996). Risks in the Use of Information Technology Within Organizations. *International Journal of Information Management*. 16(1). 65-74.

Duncombe R., Heeks R. (1999). Information, ICTs and Small Enterprise: Findings from Botswana. *Development Informatics*. Working Paper Series No. 7. Manchester: Institute for Development Policy and Management. (15.10.2000) http://idpm.man.ac.uk/idpm/di_wp7.htm

Easterby-Smith M., Thorpe R., Lowe A. (1991). *Management Research: An Introduction*. London: Sage.

Ehn P. (1988). *Work-Oriented Design of Computer Artifacts*. Ph.D. Diss. Arbetslivscentrum. Stockholm.

Ehn P. (1989). *Work Oriented Design of Computer Artifacts*. Hillsdale. NJ: Lawrence Erlbaum.

Ehn P., Sandberg Å. (1979). *Management Control and Wage Earner Power* (Foretagsstyrning och Lontagarmakt). Falkoping: Prisma.

Ein-Dor P., Segev E. (1993). A Classification of Information Systems: Analysis and Interpretation. *Information Systems Research.* 4(2). 166-204.

Eisenhardt K.C., Schoovenhoven C.B. (1996). Resource-based view of strategic alliance formation: strategic and social effects in enterpreneurial firms. *Organization Science.* 7(2). 136-150.

Engeström Y. (1987a). *Learning by expanding*. Helsinki: Orienta-konsultit.

Engeström Y. (1987b). Uusi oppimisnäkemys ja kehittyvä työntutkimus (The new view on learning and the developing work research). In Finnish. Lectio praecursoria. *Aikuiskasvatus* 7(4). 170-173.

Engeström Y. (1990a). *Activity Theory and individual and social transformation*. Opening address at the 2nd Intl. Congress for Research on Activity Theory. Lahti, Finland.

Engeström Y. (1990b). *Learning, working and imagining, twelve studies in activity theory*. Helsinki: Orienta-Konsultit Oy.

Engeström Y. (1991). Developmental Work Research: Reconstructing expertise through expansive learning. In Nurminen, Weir (eds.). *Human Jobs and Computer Interfaces.* Proceedings of the IFIP WG 9.1 Working Conference. Elsevier. 265-290.

Engeström Y. (1999). Activity theory and individual and social transformation. In Engeström, Miettinen and Punamäki (eds.). *Perspectives on Activity Theory*. Cambridge: Cambridge University Press. 1938-16.

Engeström Y., Brown K., Engeström R., Koistinen K. (1990). Organizational forgetting: an Activity-Theoretical perspective. In Middleton and Edwards (eds.). *Collective Remembering*. SAGE Publications. 139-168.

Engeström Y., Engeström R. (1986). Developmental Work Research: The Approach and an Application in Cleaning Work. *Nordisk Pedagogik.* 1. 2-15.

Engeström Y., Engeström R., Saarelma O. (1988). Computerizing medical records, production pressures and compartmentalization in the work activity of health center physicians. In *Proceedings of the Conference on Computer-Supported Cooperative Work*. Portland. Oregon. USA: ACM Press. 65-84.

Engeström Y., Escalante V. (1996). Mundane Tool or Object of Affection? The Rise and Fall of the Postal Buddy. In Nardi (ed.). *Context and Consciousness: Activity Theory and Human-Computer Interaction*. Cambridge MA: MIT Press. 325-373.

Engeström Y., Miettinen R. (1999). Introduction. In Engeström, Miettinen and Punamäki (eds.). *Perspectives on Activity Theory*. Cambridge: Cambridge University Press. 1-16

Eriksson I., Nurminen M. (1991). Doing by Learning: Embedded Application Systems. *Journal of Organizational Computing* 1(4). 323-339.

Escobar A. (1992). Reflection on 'Development': grassroots approaches and alternative politics in the Third World, *Futures* 24. 411-436.

Esteva G. (1992). Development. In: Sachs (ed.). *The development dictionary.* London: Zed Books Ltd. 6-25.

Esteva G., Prakash M.S. (1998). *Grassroots Post-Modernism: Remaking the Soil of Cutures*. London: Sed Books Ltd.

Etzioni A. (1968). *The active society: Theory of societal and political processes*. New York: The Free Press.

Fairley R. (1994). Risk management for software projects. *IEEE Software*. May. 57-67.

Falola T. (1999). *The history of Nigeria*. London: Greenwood Press.

Fitzgerald G., Hirschheim R.A., Mumford E., Wood-Harper A.T. (1985). Information Systems Research Methodology: An Introduction to the Depate. In Mumford, Hirschheim, Fitzgerald and Wood-Harper (eds.). *Research Methods in Information Systems*. North-Holland. Amsterdam: Elsevier. 3-9.

Fitzpatrick G. (2000). Centres, Peripheries and Electronic Communication: Changing Work Practice Boundaries. *Scandinavian Journal of Information Systems.* 12. 115-148.

Friedman A.L. (1989). *Computer Systems Development: History, Organization and Implementation*. West Sussex: John Wiley & Sons.

Galliers R. (1985). In Search of a Paradigm for Information Systems Research. In Mumford, Hirschheim, Fitzgerald and Wood-Harper (eds.). *Research Methods in Information Systems*. Norh-Holland. Amsterdam: Elsevier. 281-297.

Galliers R. (1991). Choosing Appropriate Information Systems Research Approahces: A Revised Taxonomy. In Nissen, Klein and Hirschheim (eds.). *Information Systems Researach: Contemporary Approaches and Emergent Traditions*. North-Holland, Amsterdam: Elsevier Science Publishers. 237-255.

Galliers R. (1994). *Information Systems Research, issues, methods and practical guidelines*. Henley-on-Thames: Alfred Waller.

Galliers R. (1997). Reflection on Information Systems Research: Twelve Points of Debate. In Mingers, Stowell (eds.). *Information Systems: An Emerging Discipline?*. Berkshire: McGraw-Hill Publishing Company. 141-157.

Genuchten M. van (1991). Why is Software Late? An Empirical Study of Reasons form Delay in software Development. *IEEE Transactions on Software Engineering.* 17(6). 582-590.

Giddens A. (1979). *Central Problems In Social Theory*. London: Macmillan.

Giddens A. (1984). *The Constitution of Society*. Cambridge: Polity Press.

Giddens A. (1990). *The Consequences of Modernity*. Cambridge: Polity Press.

Glaser B. Strauss A. (1967). *The discovery of grounded theory: Strategies of qualitative research*. London: Wiedenfeld and Nicholson.

Goldkuhl G., Lyytinen K. (1982). A language action view of information system. In Ross, Ginzberg (eds.). *Proceedings of the Third Conference on Information*

Systems. Michigan: Ann Arbor. 13-29.

Goldkuhl G., Lyytinen K. (1984). Information system specification as rule reconstruction. In Bemelmans (ed.). *Beyond Productivity, Information Systems Development for Organizational Effectiveness*. North-Holland. Amsterdam: Elsevier Science Publishers. 79-94.

Greenbaum J., Kyng M. (eds.) (1991). *Design at Work: Co-operative Design and Computer Systems.* London: Lawrence Erlbaum Associates.

Grundey M., Heeks R. (1998). Romania's Hardware and Software Industry: Building IT Policy and Capabilities in a Transitional Economy. *Development Informatics*. Working Paper Series Paper no. 2. Institute for Development Policy and Management.

Habermas J. (1984). *The theory of communicative action – reason and the rationalization of society*. Vol 1. Boston MA: Beacon Press.

Habermas J. (1987). *The theory of communicative action – Volume Two: lifeworld and systems; A critique of functionalist reason*. Boston MA: Beacon Press.

Harvey L. (1997). A Discourse on Ethnography. In Lee, Liebenau and DeGross (eds.). *Information Systems and Qualitative Research*. London: Chapman & Hall. 207-224.

Hassan S.Z. (1998). A Framework for IT Industry Development: A Case Study of Pakistan. *Journal of Global Information Technology Management*. 1(4). 38-55.

Heeks R. (1996a). Building Software Industries in Africa. *Information Technology in Developing Countries*. 6 (4). 5-7.

Heeks R. (1996b). Promoting software production and export in developing countries. In Roche, Blaine (eds.); *Information Technology, Development and Policy: Theoretical Perspectives and Practical Challenges*. Aldershot: Avebury. 77-94.

Heeks R. (1999a). Information and Communication Technologies, Poverty and Development. In *Development Informatics*. Working Paper Series. Paper No. 5. Institute for Development Policy and Management

Heeks R. (1999b). Software Strategies in Developing Countries. *Communication of the ACM*. 42 (6). 15-20.

Heeks R., Mundy D., Salazar A. (1999). Why Health Care Information Systems Succeed or Fail. In *Information Systems for Public Sector Management*. Working Paper Series Paper No. 9. Institute for Development Policy and Management. Manchester. June.

Helle M. (2000). Disturbances and Contradictions as Tools for Understanding Work in the Newsroom. *Scandinavian Journal of Information Systems* 12. 81-114.

Hettne B. (1990). *Development Theory and the Three Worlds*. New York: Longman Scientific & Technical.

Hirschheim R. (1985). Information Systems Epistemology: An Historical Perspective. In Mumford, Hirschheim, Fitzgerald and Wood-Harper (eds.). *Research Methods in Information Systems*. Norh-Holland. Amsterdam: Elsevier. 13-36.

Hirschheim R, Klein K. (1989). Four Paradigms of Information Systems Development. *Communication of the ACM*. 32(10). 1199-1216.

Hirschheim R., Klein H. (1992). A research agenda for future information systems development methodologies. In Cotterman and Senn (eds.). *Challenges and strategies for research in systems development*. West Succex: John Wiley & Sons.

Hirschheim R., Klein H., Lyytinen K. (1995). *Information Systems Development and Data Modeling. Conceptual and Philosophical Foundations*. Cambridge: Cambridge University Press.

Hirschheim R., Klein H., Lyytinen K. (1996). Exploring the Intellectual Structures of Information Systems Development: A Social Action Theoretic Analysis. *Accounting, Management and Information Technology*. 6 (1/2). 1-64.

Hirschheim R., Newman M (1991). Symbolism and information systems development: myth, metaphor and magic. *Information Systems Research*. 2(1). 29-62.

Hofstede G. (1980) *Culture's Consequences: International Differences in Work-Related Values*. Beverly Hills: Sage.

Hofstede G. (1991) *Cultures and Organisations: Software of the Mind*. London: McGraw-Hill.

Holbæk-Hanssen E., Håndlykken P., Nygaard K. (1976). *System Description and the DELTA Language*. Oslo: The Norwegian Computing Center. Publication 523.

Hopkins A.G. (1973). *Economic History of West Africa*. New York: Longman Group Ltd.

Hyysalo S., Lehenkari J. (2001). An Activity-Theoretical Method for Studying Dynamics of User-Participation in IS Design. In Anders, Mørch and Opdahl (eds.). *Electronic proceedings of 24th Information Systems Research Seminar in Scandinavia (IRIS24)*, Norway.

Isichei E. (1983). *A history of Nigeria*. London: Longman Group Ltd.

Iivari J. (1991). Contemporary schools of information systems development. *European Journal of Information Systems* 1. 249-272.

Iivari J., Hirschheim R., Klein H.K. (2001). A Dynamic Framework for Classifying Information Systems Development Methodologies and Approaches. *Journal of Management Information Systems*. 17(3). 179-218.

Iivari J., Lyytinen K. (1998). Research in Information Systems Development in Scandinavia – Unity in Plurality. *Scandinavian Journal of Information Systems* 10(1&2). 135-185.

Iliffe J. (1995). *Africans, the history of continent*. Cambridge: Cambridge University Press.

Israel J. (1979). *The Language of Dialectics and the Dialectics of Language*. Copenhagen: Munksgård.

Jacobson I., Christerson M., Jonsson P., Övergaard G. (1992). *Object-Oriented Software Engineering, A use case driven approach*. Wokingham. England: Addison Wesley.

Jacobson I., Ericsson M., Jacobson A. (1995). *The Object Advantage, Business Process Reengineering with Object Technology*. Wokingham: Addison-Wesley.

Jaiyesimi S.B. (1990). *The Nigerian situation*. Paper presented in the (Commonwealth Secreteriat) Information Technology Policy Workshop. United Kingdom, 12-16 November: Mimeo.

Jensen M. (1999). *The Status of African Information Infrastructure*. Economic

Commission for Africa. First Meeting of the Committee on Development Information (CODI). Addis Ababa. (30.11.2001) http://www.bellanet.org/partners/AISI/codipap1.htm.

Jones M. (1997). It All Depends What You Mean by Discipline. In Mingers, Stowell (eds.). *Information Systems: An Emerging Discipline?*. Berkshire: McGraw-Hill Publishing Company. 97-111.

Judd C.M., Smith E.R., Kidder L.H. (1991). *Research Methods in Social Relations*. Toronto: Holt. Rinehart and Winston.

Järvelä M., Kuvaja K. (2001). Environment, Health and Information Activities for Communities in Africa - ENHICA. In Järvelä, Kuvaja, Korpela (eds.). *African flows, Interim Report of the Research Network – ENHICA*. Working Papers no 107. Dept. of Social Sciences and Philosophy. Jyväskylä: Jyväskylä University Printing House.19-36.

Järvenpää S., Leidner D. (1998). An Information Company in Mexico: Extending the Resource-Based View of the Firm to a Developing Country Context. *Information Systems Research*. 9(4). 342-361.

Järvinen P. (1999). *On Research Methods*. Tampere: Juvenes-Print.

Karsten E. (2000). *Weaving Tapestry: Collaborative Information Technology and Organizational Change*. Doctoral Dissertation. University of Jyväskylä: Jyväskylä Studies in Computing 3.

Keen P.G.W. (1981). Information Systems and Organizational Change. *Communications of the ACM*. 24(1). 24-33.

Keil M., Cule P., Lyytinen K., Schmidt R. (1998). A Framework for Identifying Software Project Risks. *Communication of the ACM* 41(11). 76-83.

Kensing F., Blomberg J. (1998). Participatory Design: Issues and concerns. *Computer Supported Cooperative Work*. 7(3-4). 167-185.

Kensing F, Simonsen J. and Bødker K. (1998). MUST – a method for Participatory Design. *Human-Computer Interaction*. 13(2). 167-198.

Kiangi G.E., Tjipangandjara K. (1996). Opportunities for information technology in enhancing socio-economic development of a developing country. In Odedra-Straub (ed.). *Global Information technology and Socio-economic Development*. Nashua: Ivy League. 73-81.

Kiggundu. M.N. (1989). *Managing Organizations in Developing Countries: An Operational and Strategic Approach*. West Hartford Connecticut: Kumarian Press.

Kitching G. (1982). *Development and underdevelopment in historical perspective: populism, nationalism and industrialization*. London and New York: Methuen.

Klein H.K., Hirschheim R., Nissen H-E. (1991). A Pluralist Perspective of the Information Systems Research Arena. In Nissen, Klein, Hirschheim (eds.). *Information Systems Research: Contemporary Approaches & Emergent Traditions*. North-Holland.Amsterdam: Elsevier Science Publishers. 1-20.

Klein H.K., Myers M.D. (1995). *The quality of Interpretive Research in Information Systems*. Department of Management Science and Information Systems. Working Paper Series. No. 89. Auckland: University of Auckland.

Kluzer S. (1990). Computer diffusion in Black Africa. A preliminary assessment. In Bhatnagar, Bjørn-Andersen (eds.). *Information Technology in Developing*

Countries. North-Holland. Amsterdam: Elsevier Scinece Publishers. 175-187.

Kontio J. (2001). *Software Engineering Risk Management, A Method, Improvement Framework, and Empirical Evaluation*. Doctoral Dissertation. Helsinki University of Technology.

Korpela M. (1990). The role of information systems ergonomics in developing countries. In Bhatnagar, Bjørn-Andersen (eds.). *Information Technology in Developing Countries*. North-Holland. Amsterdam: Elsevier Scinece Publishers. 265-275.

Korpela M. (1992). Health Informatics in Africa: Ethics and Appropriateness. In: Bhatnagar and Odedra (eds.). *Social Implications of Computers in Developing Countries.* New Delhi: Tata McGraw-Hill. 285-294.

Korpela M. (1994). *Nigerian Practice in Computer Systems Development: A Multidisciplinary Theoretical Framework, Applied to Health Informatics*. Dissertation. Helsinki University of Technology.

Korpela M. (1995). Who Will Implement Information Technology for Development?. *South African Computer Journal* 15. 20 – 25.

Korpela M. (1996). Computer systems development for 'delinking' in Nigeria?. In Odedra-Straub (ed.). *Global Information Technology and Socio-economic Development*. New Hampshire: IVY League. 116-129.

Korpela M. (1999). Activity analysis and development in a nutshell. Handout. (15.10.2001) http://www.uku.fi/atkk/actad/nutshell.html.

Korpela M., Eerola A., Mursu A., Soriyan H.A. (2001a). Use Cases as Actions within Activities: Bridging the Gap Between Information Systems Development and Software Engineering. Abstract in: *2nd Nordic-Baltic Conference on Activity Theory and Sociocultural Research*. Sweden. p. 51.

Korpela M., Mursu A., Soriyan H.A. (2001b). Two Times Four integrative levels of analysis: A framework. In Russo, Fitzgerald, DeGross (eds.). *Realigning Research and Practice in Information Systems Development: The Social and Organisational Perspective*. IFIP WG 8.2 Working Conference. Boston. MA: Kluwer Academic. 367-377.

Korpela M., Mursu A., Soriyan H.A., Eerola A. (2002a). Information Systems Research and Information Systems Practice in a Network of Activities. In Floyd, Dittrich, Klischewski (eds.). *Social Thinking – Software Practice*. Cambridge. MA: MIT Press. 287-308.

Korpela M., Mursu A., Soriyan H.A., Olufokunbi K.C., (2002b). Information Systems Development as an Activity. *Computer Supported Cooperative Work*. 11(1-2), Special Issue on Activity Theory and the Practice of Design. 111-128.

Korpela M., Soriyan H.A., Olufokunbi K.C., Onayade A.A., Davies-Adetugbo A., Adesanmi D. (1998). Community Participation in Health Informatics in Africa: An Experiment in Tripartite Partnership in Ile-Ife, Nigeria. *Computer Supported Cooperative Work* 7. 339-358.

Korpela M., Soriyan H.A., Olufokunbi K.C. (2000a). Activity Analysis as a Method for Information Systems Development: General Introduction and Experiments from Nigeria and Finland. *Scandinavian Journal of Information Systems* 12. 191-210.

Korpela M, Soriyan HA, Olufokunbi KC, Mursu A (2000b). Made-in-Nigeria systems development methodologies: An action research project in the health sector. In Avgerou C, Walsham G, eds. *Information Technology in Context: Implementing Systems in the Developing World*. Aldershot: Ashgate. 134–152.

Kuutti K. (1991). Activity Theory and its applications to information systems research and development. In Nissen, Klein, Hirschheim (eds.). *Information Systems Research: Contemporary Approaches & Emergent Traditions*. North-Holland: Elsevier Science Publishers. 529-549.

Kuutti K. (1994). *Information Systems, Cooperative Work and Active Subjects: The Activity-Theoretical Perspective*. Research Papers. Series A 23. Oulu University Printing Centre.

Kuutti K. (1996). Activity Theory as a Potential Framework for Human-Computer Interaction Research. In Nardi (ed.). *Context and Consciousness: Activity Theory and Human-Computer Interaction*. Cambridge MA: MIT Press. 17-44.

Kuutti K., Molin-Juustila T. (1998). Information System Support for 'Loose' Coordination in a Network Organization: an Activity-Theory Perspective. In Hasan, Gould and Hyland (eds.). *Information Systems and Activity Theory: Tools in Context*. Australia: University of Wollongong Press. 73-92.

Kyng M. (1996). *Computers and Users*. Dr. Scientific thesis. Aarhus University.

Kyng M., Greenbaum J. (1991). Cooperative Design: Bringing together the Practices of Users and Designers. In Nissen, Klein, Hirschheim (eds.). *Information Systems Research: Contemporary Approaches & Emergent Traditions*. Amsterdam: North-Holland. 405-415.

Kyng M., Mathiassen L. (1982). Systems Development and Trade Union Activities. In Bjørn-Andersen (ed.). *Information Society, for Richer, for Poorer*, Amsterdam: North-Holland.

Käkölä T. (1996). *Dual Information Systems in Hyperknowledge Organizations*. Ph.D. Diss. Turku Centre for Compute Science. TUCS Dissertation no. 2. Turku.

Känsälä K. (1997). Integrating Risk Assessment with Cost Estimation. *IEEE Software*. May/June. 61-67.

Lai L.S.L. (1996). A pluralistic approach to information systems development: What can it offer to the developing world?. In Odedra-Straub (ed.). *Global Information Technology and Socio-economic Development*. New Hampshire: IVY League. 161-172.

Land F. (1994). The Information Systems Domain. In Galliers (ed.). *Information Systems Research, issues, methods and practical guidelines*. Henley-on-Thames: Alfred Waller.

Langefors B. (1966). *Theoretical Analysis of Information Systems*. Lund. Sweden: Studentlitteratur.

Langefors B. (1974). Information Systems. In *Information Processing 74*. Amsterdam: North-Holland. 937-945.

Lanzara G. Mathiassen L. (1985). Mapping situations within a system development project. *Information & Management* 8. 3-12.

Latour B. (1987). *Science in Action: How to Follow Scientists and Engineers through Society*. Cambridge. Massachusetts: Harvard University Press.

Leavitt H.J. (1964). Applied Organization Change in Industry: Structural, Technical, and Human Approaches. In Leavitt (ed.). *New Perspectives in Organizational Research*. Chichester: John Wiley. 55-71.

Lee A.S., Liebenau J. (1997). Information Systems and Qualitative Research. In Lee, Liebenau, DeGross (eds.). *Information Systems and Qualitative Research*. London: Chapman & Hall. 1-8.

Leontjev A.N. (1978). *Activity, Consciousness and Personality*. Englewood Cliffs. NJ: Prentice Hall.

Leppänen A. (2000). *Työprosessien mallintaminen tukemaan työn ja osaamisen kehittymistä*. Helsinki: Työterveyslaitos.

Leys C. (1982). Samuel Huntington and the end of classical modernization theory. In: Alavi, Shanin (eds.). Introduction to the Sociology of ”Developing Societies”. Macmillan. 332-349.

Lind P. (1996). Good use and misuse of analysis applications for decision support in enterprises. In Odedra-Straub (ed.). *Global Information technology and Socio-economic Development*. Nashua: Ivy League. 35-.

Lopez E.J., Vilaseca M.G. (1996). IT as a global economic development tool. In Roche, Blaine (eds.). *Information Technology, Development and Policy: Theoretical Perspectives and Practical Challenges*. Aldershot: Avebury. 57-75.

Lucas H.C. Jr. (1975). *Why Information Systems Fail.* New York: Columbia University Press.

Lummis C.D. (1992). Equality. In Sachs (ed.). *The development dictionary*. London: Zed Books Ltd. 38-52.

Luneberg M. (1976). *Some Propositions Concerning Analysis and Design of Information Systems.* Ph.D. Diss. Department of Information Processing Science. TRITA-IBDB-4080. Stockholm: The Royal Institute of Technology and the University of Stockholm.

Luneberg M., Andersen E.S. (1974). *Systemering, Information analys*. Lund: Studentlitteratur.

Luria A.R. (1976). *Cognitive development: Its cultural and social foundations*. Cambridge, MA: Hardvard University Press.

Lyytinen K. (1987). A Taxonomic Perspective of Information Systems Development. Theoretical Constructs and Recommendations. In Boland et al. (eds.). *Critical Issue in Information Systems Research*. Chichester: John Wiley.

Lyytinen K. (1988). Expectation Failure Concept and Systems Analysts' View of Information System Failures: Results of an Exploratory Study. *Information & Management*. 14. 45-56.

Lyytinen K. (1989). New Challenges of Systems Development. A Vision of the 90's. *Data Base*. Fall.

Lyytinen K., Hirschheim R.(1987). *Information Systems Failures - a survey and classification of the empirical literature*. Oxford Surveys in Information Technology. Oxford University Press. Vol 4. 257-309.

Lyytinen K., Klein H. (1985). The Critical Theory of Jurgen Habermas as a Basis for a Theory of Information Systems. In Mumford, Hirschheim, Fitzgerald, Wood-Harper (eds.). *Research Methods in Information Systems*. North-Holland: Elsevier Science Publishers. 219-236.

Lyytinen K. Mathiassen L., Ropponen J. (1996). A Framework for Software Risk Management. *Journal of Information Technology* 11(4). 275-285.

Lyytinen K., Mathiassen L., Ropponen J. (1998). Attention Shaping and Software Risk – A Categorical Analysis of Four Classical Approaches. *Information Systems Research* 9(3). 233-255.

Lyytinen K., Ngwenyama O. (1992). What does computer support for cooperative work mean? A structurational analysis of computer supported cooperative work. *Accounting, Management and Information Technologies*. 2(1). 19-37.

Lyytinen K., Robey D. (1999). Learning failure in information systems development. *Information Systems Journal*. 9. 85-101.

Lyytinen K., Rose G., Welke R. (1998). The brave new world of development in the internetwork computing architecture (InterNCA): or how distributed computing platforms will change systems development. *Information System Journal* 8(3). 241-253.

Machias-Chapula C.A., Alcantar J., Cruz-Aguilar B. (1998). Development of a technology information (intelligence) system in Mexico, a case study. In Proceedings of IFIP WG 9.4 conference. Bangkok. *Implementation and evaluation of information systems in developing countries*. 178-187.

Madon S. (1994). Are Methodologies Exportable to Developing Countries?. *Newsletter of IFIP Working Group 9.4 and Commonwealth Network for Information Technology*. 4 (3). 10-11.

Madsen O.L., Møller-Pedersen B., Nygaard K. (1993). *Object-Oriented Programming in the BETA Programming Language*. Wokingham: Addison-Wesley.

March J., Shapira Z. (1987). Managerial perspectives on risk and risk taking. *Management Science* 33(11). 1404-1418.

Margetts H. (1994). Informatization in public sector organizations: Distinctive or common risks?. *Information and the Public Sector*. 3 (1). 1-19.

Markus M.L. (1983). Power, Politics, and MIS Implementation. *Communication of the ACM.* 26(6). 430-444.

Markus M.L. (2000). Toward an integrated theory of IT-related risk control. In Baskerville, Stage, DeGross (eds.). *Orgnizational and Social Perspectives on Information Technology*. IFIP TC8 WG 8.2. Boston: Kluwer Academic Publishers. 167-178.

Marschak J. (1974). *Economic Information, Decision and Prediction: Selected essays*. Volume II. Dordrecht.

Marsh C. (1982). *The Survey Method*. London: George Allen & Unwin.

Martin J. (1982). *Strategic Data Planning Methodologies*. New Jersey: Prentice Hall.

Mathiassen L. (1998a). Reflective Systems Development. *Scandinavian Journal of Information Systems.* 10(1&2). 67-118.

Mathiassen L. (1998b). *Reflective Systems Development*. Dr. Technical Thesis, Aalborg University.

Mathiassen L. (2000). Collaborative practice research. In Baskerville, Stage, DeGross (eds.). *Orgnizational and Social Perspectives on Information Technology*. IFIP TC8 WG 8.2. Boston: Kluwer Academic Publishers. 127-148.

Mazrui A.A. (ed.) (1999). *General history of Africa, VIII Africa since 1935*. UNESCO. California: James Currey Ltd.

McFarlan F.W. (1981). Portfolio approach to information systems. *Harvard Business Review* 59(5). 142-150.

McGuire W.J. (1984). A contextualist theory of knowledge: Its implications for innovation and reform in psychological research. In Berkowitz (ed.). *Advances in Experimental Social Psychology*. 16. New York: Academic.

Mingers J., Stowell F. (eds.) (1997). *Information Systems: An Emerging Discipline?*. Berkshire: McGraw-Hill Publishing Company.

Mingers J. (2001). Combining IS Research Methods: Towards a Pluralist Methodology. *Information Systems Research*. 12 (3). 240-259.

Monarch I.A., Konda S.L., Carr M.J. (1996). Software Engineering Risk Repository. In *Proceedings of the 1996 SEPG Conference Software Engineering Institute*. Pittsburgh. PA.

Monteiro E., Hanseth O. (1996). Social Shaping of Information Infrastructure: On Being Specific About the Technology. In Orlikowski, Walsham, Jones, DeGross (eds.). Information Technology and Changes in Organizational Work. London: Chapman & Hall.

Morgan G. (1983). *Beyond Method*. Beverly Hills CA: Sage Publications.

Moynihan T. (1997). How Experienced Project Managers Assess Risk. *IEEE Software*. May/June. 35-41.

Mumford E. (1983). *Designing Human Systems: The ETHICS Method*. Manchester Business School.

Mumford E. (1991). Information Systems Research – Leaking Craft or Visionary Vehicle? In Nissen, Klein, Hirschheim (eds.). *Information Systems Research: Contemporary Approaches & Emergent Traditions* Amsterdam: North-Holland. 21-26.

Mumford E. (1995). *Effective Requirements Analysis and Systems Design: The ETHICS Method*. Basingstoke: Macmillan.

Mumford E. (2000). Socio-technical design: an unfilled promise or a future opportunity?. In Basekerville, Stage, DeGross (eds.). *Orgnizational and Social Perspectives on Information Technology*. IFIP TC8 WG 8.2. Kluwer Academic Publishers. 33-46 .

Mumford E., Hensall D. (1979). *A Participative Approach to Computer Design*. London: Associated Business Press.

Mumford E., Hirschheim R., Fitzgerald G., Wood-Harper A.T. (1985). *Research Methods in Information Systems*. North-Holland: Elsevier Science Publishers.

Mumford E., Ward T.B. (1968). *Computers: Planning for People*. London: Batsford.

Mumford E., Weir M. (1979). *Computer Systems in Work Design – the ETHICS method*. London: Associated Business Press.

Mundy D. (1996). IT in developing countries - a loss of independence?. In Roche, Blaine (eds.). *Information Technology, Development and Policy: Theoretical Perspectives and Practical Challenges*. Aldershot: Avebury. 45-56.

Munro J.F. (1976). *Africa and the International Economy 1800-1960*. London: J.M. Dent & Sons Ltd.

Mursu A., Lyytinen K., Soriyan H.A., Korpela M. (submitted). Identifying Software Project Risks in Nigeria: An International Comparative Study. *submitted to EJIS*.

Mursu A., Soriyan H.A., Korpela M. (2001). Information Systems Development in Nigerian Software Companies – A Case of Small Locally Owned Company. In Anders, Mørch and Opdahl (eds.). *Electronic proceedings of 24th Information Systems Research Seminar in Scandinavia (IRIS24)*, Norway.

Mursu A., Soriyan H.A., Olufokunbi K.C., Korpela M. (1999). Toward Successful ISD in Developing Countries: First Results from a Nigerian Risk Study Using the Delphi Method. In Käkölä T. (ed.). Proceedings of the 22nd Information Systems Research Seminar in Scandinavia (IRIS 22): "*Enterprise Architectures for Virtual Organisations*". Jyväskylä: University of Jyväskylä Computer Science and Information Systems Reports. Technical Reports. TR-21. Vol. 2. 397-414.

Mursu A., Soriyan H.A., Olufokunbi K., Korpela M. (2000). Information systems development in a developing country: Theoretical analysis of special requirements in Nigeria and Africa. In Sprague RH Jr. (ed.). *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*. Hawaii. USA. 4–7 January 2000. Los Alamitos. CA: IEEE Computer Society. 185 (full text on CD-ROM).

Myers M. (1997). Interpretive Research in Information Systems. In Mingers, Stowell (eds.). *Information Systems: An Emerging Discipline?*. Berkshire: McGraw-Hill Publishing Company. 239-265.

Myers M. (1999). Investigating Information Systems with Ethnographic Research. *Communication of the Association for Information Systems*. 2(23).

Mäkeläinen B., Nurminen M., Reijonen P., Torvinen V. (1996). Everyday Use between Success and Failure: Making Sense with ONION Layers. In Dahlbom, Ljungberg, Nuld•n, Simon, Stage and Sørensen (eds.). *Proceedings of the 19th IRIS Information Systems Research Seminar in Scandiavia, The future*. 141-153.

Nandhakumar J., Vidgen R. (2001). Due Process and the Introduction of New Technology: The Institution of Video Teleconferencing. In Russo, Fitzgerald, DeGross (eds.). *Realing Research and Practice in Information Systems Development: The Social and Organizational Perspective*. Boston MA: Kluwer Academic. 127–148.

Nardi B.A. (ed.) (1996). *Context and Consciousness: Activity Theory and Human-Computer Interaction*. Cambridge MA: MIT Press.

Nederveen Pieterse J. (2000). After post-development. *Third World Quarterly* 21(2). 175-191.

Neko Y.M. (1992). The cultural and social impact of new information technologies in Africa. In Bhatnagar, Odedra (eds.). *Social Implications of Computers in Developing Countries*. New Delhi: Tata McGraw-Hill. 63 – 69.

Neumann P. G. (1995). *Computer Related Risks*. New York: Addison-Wesley Publishing Company.

Ngwenyama O.K. (1991). The Critical Social Theory Approach to Information Systems: Problems and Challenges. In Nissen, Klein, Hirschheim (eds.). *Information Systems Research: Contemporary Approaches & Emergent Traditions.* North-Holland: Elsevier Science Publishers. 267-280.

Nicholson B., Sahay S., Krishna S. (2000). Work practices and local improvisations within global software teams: a case study of a UK subsidiary in India. In Proceedings of IFIP WG 9.4. *Information Flows, Local Improvisations and Work Practices*. Cape Town.

Nissen H-E., Klein H.K., Hirschheim R (1991). *Information Systems Research: Contemporary Approaches & Emergent Traditions*. North-Holland: Elsevier Science Publishers.

Nulens G. (1998). Information technology in Africa. The policy of the World Bank. In Avgerou (ed.). *Implementation and evaluation of information systems in developing countries*. Proceedings of the fifth IFIP WG 9.4 conference. Bangkok.

Nurminen M., Reijonen P., Tuomisto A. (1994). Whose work is software? In Bradley and Hendrick (eds.). *Human Factors in Organizational Design and Management* - IV. North-Holland: Elsevier Science. 381-386.

Nurminen M. Torvinen V. (1997). Evaluating Information Systems Deployment through Work-Roles. In Galliers, Murphy, Hansen O'Callaghan, Carlsson and Loebbecke (eds.). *Proceedings of the 5th European Conference of Information Systems. vol II*. Cork Publishing Limited. 707-720.

Nustad K.G (2001). Development: the devil we know?. *Third World Quarterly* 22(4). 479-489.

Nygaard K. (1979). The Iron and Metal Project: Trade Union Participation. In Sandberg (ed.). *Computers Dividing Man and Work – Recent Scandinavian Research on Planning and Computers from a Trade Union Perspective*. Malmø. Sweden: Swedish Center for Working Life. Demos Project report no. 13. Utbildningsproduktion.

Odedra M. (1990). Information technology in Kenya and Indonesia: a comparative analysis. In Bhatnagar, Bjørn-Andersen (eds.). *Information Technology in Developing Countries*. North-Holland: Elsevier Scinece Publishers. 189-199.

Odedra M. (1992). Much more than human resource development for Africa. In Bhatnagar (ed.). *Information technology manpower, key issues of developing country*. New Delhi: Tata McGraw-Hill. 189-199.

Odedra-Straub M. (1996). Introduction In Odedra-Straub (ed.) *Global Information Technology and Socio-economic Development*. New Hampshre: Ivy League. 1-7.

OECD (1997). Information Technology Outlook. Paris: OECD. (30.10.1999) http://www.oecd.org/dsti/sti/

Ojo S.O. (1992). Socio-Cultural and Organisational Issues in IT Applications in Nigeria. In Bhatnagar, Odedra (eds.). *Social Implications of Computers in Developing Countries*. New Delhi: Tata McGraw-Hill. 99-109 .

Okot-Uma R. W'O (1992). A perspective of contextual, operational and strategy: problems of information in developing countries. In Bhatnagar and Odedra (eds.). *Social Implications of Computers in Developing Countries*. New Delhi: Tata McGraw-Hill Publishing Company Ltd. 10-25.

Okuwoga O. (1990). Impact of Information Technology on Nigeria's Socio-Economic Development. In Bhatnagar, Bjørn-Andersen (eds.). *Information Technology in Developing Countries*. North-Holland: Elsevier Scinece Publishers. 101-148.

Oliver R., Atmore A. (1981). *The African Middle Ages 1400-1800*. Cambridge: Cambridge University Press.

Olivier A.P.S. (1996). An information technology policy towards development in a developing country: The situation in South Africa. In Odedra-Straub (ed.). *Global Information Technology and Socio-economic Development* New Hampshire: IVY League. 181-191.

Orlikowski W.J. (1995). *Action and Artifact: The structuring of Technologies-in-Use*. Sloan Schoold of Managemetn. Cambridge. MA: Sloan School of Management Working Paper.

Orlikowski W.J., Baroudi J. (1991). Studying Information Technology in Organizations: Research Approaches and Assumptions. *Information Systems Research* 2(1). 1-28.

Oyomno G. Z. (1996). Sustainability of governmental use of microcomputer-based information technology in Kenya. In Odedra-Straub (ed.). *Global Information technology and Socio-economic Development*. Nashua: Ivy League. 19-34.

Panos. (1998). *The Internet and Poverty: Real Hope or Real Hype?*. London: Panos.

Pellegrini U. (1980). The problem of appropriate technology. In De Giorgio, Roveda (eds.). *Criteria for Selecting Appropriate under Different Cultural, Technical and Social Conditions*. Proceedings of The IFAC Symposium. Italy: Pergamon Press. 1-5.

Perrow C. (1984). *Normal Accidents: Living with High-Risk Technologies*. New York: Basic Books.

Pervan G.P., Akaphant S. (1998). A Survey of Key IS/IT Management Issues in the Thailand Public Sector. In Avgerou C. (ed.). *Implementation and evaluation of information systems in developing countries*. Vienna: International Federation for Information Processing. 372-388.

Pettigrew A. (1985). Contextualist Research and the Study of Organizational Change Processes. In Mumford, Hirschheim, Fitzgerald, Wood-Harper (eds.). *Research Methods in Information Systems*. North-Holland: Elsevier Science Publishers. 53-78.

Poulymenakou A., Holmes A. (1996). A contingency framework for the investigation of information sytems failures. *European Journal of Information Systems* 5. 34-46.

Preston P. (1986). *Making Sense of Development*. New York.

Probert S. (1997). The actuality of information systems. In Mingers, Stowell (eds.). *Information Systems: An Emerging Discipline?*. Berkshire: McGraw-Hill Publishing Company. 21-62.

Rathswohl E. (1991). Applying Don Ihde's Phenomenology of Instrumentation as a Framework for Designing Research in Information Science. In Nissen, Klein, Hirschheim (eds.). *Information Systems Research: Contemporary Approaches & Emergent Traditions*. North-Holland: Elsevier Science Publishers. 421-438.

Redclift M. (1992). *Sustainable Development: Exploring the Contradictions*. London: Routledge.

Reenskaug T., Wold P., Lehne O.A. (1995). *Working with Objects, The OOram software engineering method*. Greenwich CT: Manning Publications.

Reynolds J., Stinson W. (1993). Sustainability Analysis. *Primary Health Care*

Management Advancement Programme. Bangkok: Aga Khan Foundation.

Robey D. (1996). Research Commentary: Diversity in Information Systems Research: Threat, Promise, and Responsibility. *Information Systems Research*. 7(4). 400-408.

Roche E.M., Blaine M.J. (1996). Introduction - information technology, development and policy. In Roche, Blaine (eds.); *Information Technology, Development and Policy: Theoretical Perspectives and Practical Challenges*. Aldershot: Avebury. 1-24.

Ropponen J. (1993). *Risk management in information system development projects*. Licentiate thesis in Information Systems. University of Jyväskylä

Ropponen J. (1999). *Software Risk Management – Foundation, Principles and Empirical Findings*. Doctoral Dissertation. University of Jyväskylä: Jyväskylä Studies in Computing 1.

Ropponen J., Lyytinen K. (1997). Can software risk management improve system development: an exploratory study. *European Journal of Information Systems* 6(1). 41-50.

Rostow W.W. (1960). *The stages of economic growth*. Cambridge: Cambridge University Press.

Rowe W. D. (1988). *An Anatomy of Risk*. Malabar. FL: Robert E. Krieger Publishing Co.

Russo N. (2000). Expanding the horizons of information systems development. In Baskerville, Stage, DeGross (eds.). *Orgnizational and Social Perspectives on Information Technology*. IFIP TC8 WG 8.2. Kluwer Academic Publishers. 103-111.

Russo N., Fitzgerald B. (2001). Directions in Information Systems Development: Integrating New Technologies, Research Approaches, and Development Practices. In Russo, Fitzgerald, DeGross (eds.). *Realing Research and Practice in Information Systems Development: The Social and Organizational Perspectives*. Boston MA: Kluwer Academic. 1-8.

Van Ryckeghem D. (1992). Implementation of information technology: socio-cultural issues. In Bhatnagar, Odedra (eds.). *Social Implications of Computers in Developing Countries*. New Delhi: Tata McGraw-Hill. 42 – 50.

Sadare D. (1998). Making Information Technology Systems work in Nigeria against all odds. In Avgerou (ed.). *Implementation and evaluation of information systems in developing countries*. Proceedings of the fifth IFIP WG 9.4 conference. 311-330.

Sandberg A. (1985). Socio-Technical Design, Trade Union Strategies and Action Research. In Mumford, Hirschheim, Fitzgerald, Wood-Harper (eds.). *Research Methods in Information Systems*. North-Holland: Elsevier Science Publishers. 79-92.

Schmidt, Bannon (1992). Taking CSCW Seriously: Supporting Articulation Work. *Computer Supported Cooperative Work* 1(1-2). 7-40.

Schmidt R. (1997). Managing Delphi surveys using nonparametric statistical techniques. *Decision Science*. 28(3). 763-774.

Schmidt R., Lyytinen K., Keil M., Cule P. (2001). Identifying Software Project Risks: An International Delphi Study. *Journal of Management Information Systems* 17(4). 5-36.

Schön D.A. (1983). *The Reflective Practitioner. How Professionals Think in Action*. New York: Basic Books.

Siegel S., Castellan N.J. (1988). *Nonparametric Statistics for the Behavioral Sciences*. McGraw-Hill.

Simons R. (1999). How Risky is your Company?. *Harvard Business Review* 77(3). 85-94.

Sisti F.J., Joseph S. (1994). *Software Risk Evaluation Method Version 1.0*, Software Engineering Institute, Pittsburgh. CMU/SEI-94-TR-19.

Smith H.A., McKeen J.D., Staples D.S. (2001). Risk Management in Information Systems: Problems and Potential. *Communications of AIS* 7 (13).

Soriyan H.A., Korpela M., Mursu A., Kailou K., Olufokunbi K. (1999). Information Systems Development for Healthcare in Africa: The INDEHELA-Methods Project. Paper presented in the 3rd *International Working Conference on Health Informatics in Africa. HELINA'99*. Zimbabwe.

Soriyan H.A., Korpela M., Mursu A. (2002). Information Systems Development in Nigerian Software Companies: The Industry Profile. accepted to *IFIP WG 9.4 conference* in Bangalore. India. 2002.

Soriyan H.A., Mursu A., Adegaye A.O., Korpela M., (2000). Information systems development in Nigerian software companies: Methodological issues and empirical findings. In: IFIP WG 9.4 Working Conference 2000: *Information Flows, Local Improvisations and Work Practices*. South Africa.

Soriyan H.A., Mursu A., Akinde A.D., Korpela M. (2001). Information systems development in Nigerian software companies: Research methodology and assessment from the healthcare sector's perspective. *Electronic Journal on Information Systems in Developing Countries*. 5(4).

Spasser M. (2000). Articulating Collaborative Activity: Design-in-use of Collaborative Publishing Services in the Flora of North America Project. *Scandinavian Journal of Information Systems*. 12. 149-172.

Spinuzzi C. (1999). Grappling with Distributed Usability: A Cultural-Historical Examination of Documentation Genres Over Four Decades. In *Proceedings of the 17th annual international conference on Computer documentation*. New Orleans. USA: ACM Digital Library.

Star S.L. (1989). The structure of ill-structured solutions: boundary object and heterogeneous distributed problem solving. In Gasser, Huhns (eds.). *Distributed artificial intelligence*. Vol. 2. London: Pitman. 37-54.

Srivihok A. (1998). Succesful EIS implementation in Thailand: What factors do you need to contribute?. In *Implementation and evaluation of information systems in developing countries*. Proceedings of IFIP WG 9.4 conference. Bangkok 1998. 214-221.

Stowell F., Mingers J. (1997). Information Systems: An Emerging Discipline? – Introduction. In Mingers, Stowell (eds.). *Information Systems: An Emerging Discipline?*. Berkshire: McGraw-Hill Publishing Company. 1-15.

Strauss A. (1993). *Continual Permutations of Action*. New York: Aldine de Gruyter.

Sundgren B. (1973). *Infological Approach to Data Bases*. Ph.D. Diss.. Skriftserie utgiven an statistiska centralbyron. Nummer 7. Stockholm: Statistiska

Centralbyron.

Thorsrud E., Emery F. (1969). *Mot en ny bedriftsorganisasjon – eksperimenter I industrielt demokrati*.. Oslo: Tanum

Toikka K., Hyötyläinen R., Norros L. (1986). Development of work in flexible manufacturing. *Nordisk Pedagogik* 6(1). 16-25.

Toraskar K. (1991). How Managerial Users Evaluate their Decision-Support?: A Grounded Theory Approach. In Nissen, Klein, Hirschheim (eds.). *Information Systems Research: Contemporary Approaches & Emergent Traditions*. North-Holland: Elsevier Science Publishers. 195-225.

Truex D.P., Baskerville R., Klein H. (1999). Growing Systems in Emergent Organizations. *Communication of the ACM* 42(8). 117-123.

Ullrich O. (1992). Technology. In Sachs (ed.). *The Development Dictionary*. London: Zed Books Ltd. 275-287.

UNDP (1997). UNDP and the communication revolution. (30.10.1999) http://www.undp.org:80/comm/index.html

UNDP (2001). Human Development Report 2001. (30.1.2002) http://www.undp.org/hdr2001/

UNIDO (1989). *Computers for Industrial Management in Africa: The case of Nigeria*. Prepared by Michal Nwachuku. United Nations Industrial Development Organization. V.89-57624. PPD.126.

United Nations (1973). *The Application of Computer Technology for Development: Second report of the Secretary-General*. UN Publication no. E.73.IIA.12

Van Scoy Roger L. (1992). *Software Development Risk: Opportunity, Not Problem*. Technical Report. CMU/SEI-92-TR-30. ESDC-TR92-030. September.

Vartiainen M., Tiitinen H., Teikari V. (1991). Interfaces in a flexible manufacturing system – a case study. In Nurminen and Weir (eds.). *Human Jobs and Computer Interfaces*. Proceedings of the IFIP WG 9.1. Elsevier. 35-46.

Vidgen R., Braa K. (1997). Balancing Interpretation and Intervention in Information Systems Research: The Action Case Approach. In Lee, Liebenau, DeGross (eds.). *Information Systems and Qualitative Research*. London: Chapman & Hall.

Vrazalic L., Gould E. (2001). Towards an Activity-based Usability Evaluation Methodology. In Anders, Mørch and Opdahl (eds.). *Electronic proceedings of 24th Information Systems Research Seminar in Scandinavia (IRIS24)*, Norway.

Vygotsky L.S. (1978). *Mind in Society: The Development of Higher Psychological Processes*. Compiled from several sources and edited by Cole M., John-Steiner V. and Scribner S. Harvard MA: Harvard University Press.

Waema T. M. (1996). Implementation of information technology projects and economic development: Issues, problems and strategies. In Odedra-Straub (ed.) *Global information technology and socio-economic development*. New Hampshire: Ivy League. 106-115.

Walsham G. (1992). Decentralization of information systems in DCs: Power to people? In Bhatnagar and Odedra (eds.). *Social Implications of Computers in Developing Countries.* Tata McGraw-Hill. New Delhi. 197-208.

Walsham G. (1993). *Interpreting Information Systems in Organisations*. New York: Wiley.

Walsham G. (1997). Actor-Network Theory and IS Research: Current Status and

Future Prospects. In Lee, Liebenau and DeGross (eds.). *Information Systems and Qualitative Research.* London: Chapman & Hall. 466-480.

Walsham G. (2000). Globalization and IT: Agenda for Research. In Baskerville, Stage and DeGross (eds.). *Organizational and Social Perspectives on Information Technology*. Boston: Kluwer Academic Publishers. 195-210.

Walsham G. (2001). *Making a World of Difference – IT in a Global Context*. Chichester: John Wiley & Sons Ltd.

Walsham G., Sahay S. (1996). *GIS for District-Level Administration in India: Problems and Opportunities*. Lancaster. United Kingdom: University of Lancaster Research Paper SM 02/96.

Walsham G., Symons V., Waema T. (1990). Information Systems as Social Systems: Implications for Developing Countries. In Bhatnagar, Bjørn-Andersen (eds.). *Information Technology in Developing Countries.* North-Holland. Amsterdam: Elsevier Scinece Publishers. 51-61.

Watts S.H. (1995). *A Discipline for Software Engineering*. Reading. MA: Addison-Wesley.

Waylen G. (1996). *The Third World and Development. In Gender in Third World Politics*. Buckingham: Open University Press.

Webster's Dictionary, *Random House Webster's Unabridged Dictionary*, Second ed. New Yourk: Random House.

Weick K. (1985). Theoretical Assumptions and Research Methodology Selection. In McFarlan (ed.). *Information Systems Research Challenge*. Harvard Business School. 111-132.

Welke R, Konsynski B. (1982). An examination of the interaction between technology, methodology and information systems: A tripartite view. *Data Base* 14. 41-57.

Wickins P. (1986). *Africa 1880-1980, An Economic History*. Cape Town: Oxford University Press.

Wicks A., Freeman R.E. (1998). Organization Studies and the New Pragmatism: Positivism. Anti-positivism, and the Search for Ethics. *Organization Science* 9(2). 123-140.

Willcocks L., Margetts H. (1994). Risk assesment and Information systems. *European Journal of Information systems* 3(2). 127-138.

Winfield I. (1991). *Organisations and Information Technology*. Oxford: Blackwell Scientific.

Wise J.A., Debons A. (eds.) (1986). Preface. In *Information Systems: Failure Analysis*. NATO ASI Series. Series F: Computer and Systems Sciences. Vol. 32. Springer-Verlag: Berlin.

Wisner A. (1990). Cognition at work, influence of cultural and technical conditions. In Noro, Brown (eds.). *Human Factors in Organizational Design and Management III*. Proceedings of the Third International Symposium, Japan. North-Holland. 445-449.

Woherem E.E. (1992). Strategy for indigenisation of information technology in Africa. In Bhatnagar, Odedra (eds.). *Social implications of computers in developing countries*. New Delhi: Tata McGraw-Hill Publishing Ltd. 71-80.

World Bank (1994). *World Development Report*. New York: Oxford University

Press.
World Bank (1999a). *World Development Report 1998/1999*. (15.10.2000) http://www.worldbank.org/html/extpb/wdr9899p.htm
World Bank (1999b). *Countries: Nigeria Country Brief*. (15.10.2000) http://www.worldbank.org/afr/ng.htm.
World Bank (2000). *World Development Report 1999/2000*. (15.10.2000) http://www.worldbank.org/wdr/2000/.
World Bank (2002). *World Development Indicators*. (14.4.2002) http://www.worldbank.org/data/wdi2002/index.htm
Yin R.K (1984). *Case Study Research, Design and Methods*. Beverly Hills. California: Sage Publications
Zwangobani E. (1990). Information technology in Africa – an overview. In Bhatnagar, Bjørn-Andersen (eds.). *Information Technology in Developing Countries*. North-Holland: Elsevier Scinece Publishers B.W. 167-173.

APPENDIX 1

TABLE 29 The list of publication by INDEHELA-Methods project

1. Korpela M., Soriyan H.A., Olufokunbi K.C., Mursu A. (1998). Blueprint for an African systems development methodology: An action research project in the health sector. In Avgerou C (ed.). *Implementation and Evaluation of Information Systems in Developing Countries. Proceedings of the Fifth International Working Conference of IFIP WG 9.4*, Bangkok, Thailand. Vienna: International Federation for Information Processing. 273-285.
(Contribution: co-author, commentator)

2. Mursu A., Soriyan H.A., Olufokunbi K.C., Korpela M. (1999). From software risks to sustainable information systems: Setting the stage for a Delphi study in Nigeria. *Journal of Global Information Technology Management* 2(3), July. 57-71.
(Contribution: main author)

3. Mursu A., Soriyan H.A., Olufokunbi K.C., Korpela M. (1999). Toward Successful ISD in Developing Countries: First Results from a Nigerian Risk Study Using the Delphi Method. In Käkölä T (ed.). *Enterprise Architectures for Virtual organizations. Proceedings of the 22nd Information Systems Research Seminar in Scandinavia (IRIS 22).* Keuruu, Finland. Jyväskylä: University of Jyväskylä, Computer Science and Information Systems Reports. Technical Reports, TR-21. Vol. 2. 397-414.
(Contribution: main author)

4. Mursu A. (1999). Risks in Information System Development in a Developing Country - a research plan for a risk study in Nigeria using Delphi Method. In Seppänen M. (ed.). *Yearbook of the Finnish Society for Development Studies, Multidiciplinarity in Development Studies*. VII, Vol. 1998-99. Helsinki: Finnish Society for Development Studies. 1-22.

5. Mursu A., Soriyan H.A., Olufokunbi K., Korpela M. (2000). Information systems development in a developing country: Theoretical analysis of special requirements in Nigeria and Africa. In Sprague R. (ed.). *Proceedings of the HICSS-33 Minitrack on 'Information Technology in Developing Countries'*. Hawaii. Abstracts and CD-ROM of Full Papers. Los Alamitos, California: IEEE Computer Society.
(Contribution: main author)

6. Soriyan H.A., Mursu A., Korpela M. (2000). Information systems development methodologies: gender issues in a developing economy. In Balka E, Smith R (eds.). *Women, Work and Computerization. Charting a Course to the Future*. Prodeedings of the seventh IFIP TC9 WG9.1 International Conference on Women, Work and Computerization. Vancouver, British Columbia, Canada. Boston: Kluwer Academic. 146-154.
(Contribution: co-author, commentator)

7. Soriyan H.A., Mursu A., Adewale O.A., Korpela M. (2000). Information systems development in Nigerian software companies: Empirical findings and methodological issues. In *IFIP WG 9.4 Conference on Information Flows, Local Improvisations and Work Practices*. Cape Town, South Africa: International Federation for Information Processing.
(Contribution: co-author, case descriptions, commentator)

8. Korpela M., Soriyan H.A., Olufokunbi K.C., Mursu A. (2000). Made-in-Nigeria systems development methodologies. An action research project in the health sector. In Avgerou C, Walsham G (eds.). *Information Technology in Context: Studies from a Developing Countries Perspective*. Aldershot: Ashgate. 134-152.
(Contribution: co-author, research framework, commentator) (continues)

TABLE 29 (continues)

9. Soriyan H.A., Mursu A., Akinde A.K., Korpela M. (2001). Information systems development in Nigerian software companies: Research methodology and assessment from the healthcare sector's perspective. *Electronic Journal on Information Systems in Developing Countries*, vol 5, 2001.
(Contribution: co-author, case descriptions, commentator)

10. Mursu A., Soriyan H.A., Olufokunbi K.C., Korpela M. (2001). Software industry in Nigeria: An outline. In Järvelä M, Kuvaja K, Korpela M. (eds.). *Environment, Health, and Information Activities for Communities in Africa*. Jyväskylä: Department of Social Sciences and Philosophy. Working Papers no 107. 171-210.
(Contribution: main author)

11. Korpela M., Mursu A., Soriyan H.A. (2001). Two times four integrative levels of analysis: A framework. In Russo N, Fitzgerald B, DeGross J. (eds.). *Realigning Research and Practice in Information Systems Development: The Social and Organizational Perspective*. Boston, MA: Kluwer Academic. 367-377.
(Contribution: co-author, commentator)

12. Mursu A., Soriyan H.A., Korpela M. (2001). Information Systems Development in Nigerian Software Companies - A Case of Small Locally Owned Company. *Proceedings of the 24th Information Systems Research Seminar in Scandinavia (IRIS 24)*, Ulvik, Norway. CD-ROM of full papers. Bergen: Department of Information Science, University of Bergen.
(Contribution: main author)

13. Korpela M., Mursu A., Soriyan H.A. (2002). Information systems development as an activity. C*omputer Supported Cooperative Work* 11(1-2). Special Issue on Activity Theory and the Practice of Design. 111-128.
(Contribution: co-author, commentator)

14. Korpela M., Mursu A., Soriyan H.A., Eerola A. (2002). Information systems research and information systems practice in a network of activities. In Floyd C, Dittrich Y, Klischewski R (eds.). *Social Thinking - Software Practice*. Cambridge, MA: MIT Press. 287-308.
(Contribution: co-author, commentator)

15. Mursu A., Soriyan H.A., Korpela M. (2002). Risky business - A case study on information systems development in Nigeria. In Krishna S., Shirin M. (eds.). *IFIP WG 9.4 Conference on ICTs and Development : New Opportunities, Perspectives and Challenges*. Bangalore, India: Indian Institute of Management. 385-401.
(Contribution: main author)

16. Soriyan H.A., Korpela M., Mursu A. (2002). Information systems development in Nigerian software companies: The industry profile. In Krishna S., Shirin M. (eds.) *IFIP WG 9.4 Conference on ICTs and Development : New Opportunities, Perspectives and Challenges*. Bangalore, India: Indian Institute of Management. 276-291.
(Contribution: co-author, commentator)

17. Mursu A., Lyytinen K., Soriyan H.A., Korpela M. (submitted). Identifying Software Project Risks in Nigeria: An International Comparative Study. *Submitted to European Journal of Information Systems.*
(Contribution: main author)

(continues)

TABLE 29 (continues)

18. Soriyan H.A., Akinde A.D., Korpela M., Mursu A., Ale F., Oyetoyan T. (submitted), Preconditions for information system development for African healthcare institutions: The MINPHIS experience and a software industry survey in Nigeria. *Submitted to Meth Inf Med 2002*.
(Contribution: co-author, commentator)

19. Korpela M., Mursu A., Soriyan H.A. (2002). Activity networks in context: ActAD as a method for information systems research and practice in Nigeria. In *ISCRAT 2002*, Fifth Congress of the International Society for Cultural Research and Activity Theory, Amsterdam, 18-21 June 2002, p. (forthcoming).
(Contribution: co-author, commentator)

20. Mursu A., Korpela M. (2002). Tietotekniikka ja kehitys – näkökulmia kestävään kehitykseen. *Janus, sosiaalipolitiikan ja sosiaalityön tutkimuksen aikakauslehti*, no 2/2002.
(Contribution: main author)

APPENDIX 2

TABLE 30 Upper critical values of the chi-square distribution

	Probability of exceeding the critical value				
	0.10	0.05	0.025	0.01	0.001
1	2.706	3.841	5.024	6.635	10.828
2	4.605	5.991	7.378	9.210	13.816
3	6.251	7.815	9.348	11.345	16.266
4	7.779	9.488	11.143	13.277	18.467
5	9.236	11.070	12.833	15.086	20.515
6	10.645	12.592	14.449	16.812	22.458
7	12.017	14.067	16.013	18.475	24.322
8	13.362	15.507	17.535	20.090	26.125
9	14.684	16.919	19.023	21.666	27.877
10	15.987	18.307	20.483	23.209	29.588
11	17.275	19.675	21.920	24.725	31.264
12	18.549	21.026	23.337	26.217	32.910
13	19.812	22.362	24.736	27.688	34.528
14	21.064	23.685	26.119	29.141	36.123
15	22.307	24.996	27.488	30.578	37.697
16	23.542	26.296	28.845	32.000	39.252
17	24.769	27.587	30.191	33.409	40.790
18	25.989	28.869	31.526	34.805	42.312
19	27.204	30.144	32.852	36.191	43.820
20	28.412	31.410	34.170	37.566	45.315
21	29.615	32.671	35.479	38.932	46.797
22	30.813	33.924	36.781	40.289	48.268
23	32.007	35.172	38.076	41.638	49.728
24	33.196	36.415	39.364	42.980	51.179
25	34.382	37.652	40.646	44.314	52.620
26	35.563	38.885	41.923	45.642	54.052
27	36.741	40.113	43.195	46.963	55.476
28	37.916	41.337	44.461	48.278	56.892
29	39.087	42.557	45.722	49.588	58.301
30	40.256	43.773	46.979	50.892	59.703

APPENDIX 3

TABLE 31 Combined list of risk factors

0.	**Socio-economic context**
0.1	*Political climate in the country, including economic situation*: the poor economical state of the country does not allow for IT improvement and investment
0.2	*IT awareness in the country*: lack of proper IT exposure and policy
0.3	*Erratic and unreliable communication network*: poor communication (network, telecommunication etc.) can hinde some projects
0.4	*Energy supply*: When regular supply of power to computer systems cannot be guaranteed
0.5	*Tertiary institutions*: tertiary institutions in country today lack a lot of facilities required to prepare student for solid IT future
0.6	*Poor copyright / intellectual property right protection*
1.	**Corporate Environment**
1.1	*A climate of change in the business and organizational environment that creates instability in the project.*
1.2	*Mismatch between company culture and required business process changes needed for new system*. A mismatch between the corporate culture and the changes required by the new system.
1.3	*Projects That Are Intended to Fail*: Projects started for political reasons which carry no clear business value but serve to divert the organization's focus from actual needed change. Such projects are under-funded, not supported and are not intended to succeed. Projects have no business value and are used as diversionary tactics to avoid facing the real change needs.
1.4	*Unstable Corporate Environment:* Competitive pressures radically alter user requirements, sometimes making the entire project obsolete.
1.5	*Change in Ownership or Senior Management:* New owners and/or managers set new business direction that causes mismatch between corporate needs and project objectives.
2.	**Sponsorship/Ownership**
2.1	*Lack of Top Management Commitment to the Project*. This includes oversight by executives and visibility of their commitment, committing required resources, changing policies as needed.
2.2	*Lack of client responsibility, ownership, and buy-in of the project and its delivered system(s).*
2.3	*Failure to gain user commitment*: Laying blame for "lack of client responsibility" on the project leader rather than on the users.
2.4	*Conflict Between User Departments*: Serious differences in project goals, deliverables, design, etc., calls into question concept of shared ownership.
2.5	*Failure to get project plan approval from all parties*
2.6	*Customer's staff turnover*
3.	**Relationship Management**
3.1	*Failure to Manage End User Expectations*: Expectations determine the actual success or failure of a project. Expectations mismatched with deliverable - too high or too low - cause problems. Expectations must be correctly identified and constantly reinforced in order to avoid failure.
3.2	*Lack of Adequate User Involvement*: Functional users must actively participate in the project team and commit to their deliverables and responsibilities. User time must be dedicated to the goals of the project.
3.3	*Lack of Cooperation from Users*: Users refuse to provide requirements and/or refuse to do acceptance testing.
3.4	*Failure to Identify All Stakeholders*: Tunnel vision leads project management to ignore some key stakeholders in the project, affecting requirements definition, implementation, etc.

(continues)

TABLE 31 (continues)

3.5	*Growing Sophistication of Users Leads to Higher Expectations*: Users are more knowledgeable, have seen sophisticated applications, apply previous observations to existing project.
3.6	*Managing Multiple Relationships with Stakeholders*: Some “clients” are also “partners” in producing deliverables in other projects. Leads to confusion of roles and responsibilities.
3.7	*Lack of appropriate experience of the user representatives*: Users assigned who lack necessary knowledge of the application or the organization
3.8	*Customer’s ability to react to change*: Certain individuals do not want to conform the change because of general phobia of computers or they are worried their jobs are threatened
3.9	*Inadequate user training*: adequate training of customer is required to maintain systems.
3.10	*Negligence of agreements*: clients make illegal duplicates of packages
4.	**Project Management**
4.1	*Not Managing Change Properly:* Each project needs a process to manage change so that scope and budget are controlled. Scope creep is a function of ineffective change management and of not clearly identifying what equals success.
4.2	*Lack of Effective Project Management Skills*: Project teams are formed and the project manager does not have the power or skills to succeed. Project administration must be properly addressed.
4.3	*Lack of Effective Project Management Methodology*: The team employs no change control, no project planning or other necessary skills or processes.
4.4	*Improper Definition of Roles and Responsibilities*: Members of the project team and the organization are unclear as to their roles and responsibilities. This includes outsourcers and consultants.
4.5	*Poor or Non-Existent Control*: No sign-offs, no project tracking methodology, unaware of overall project status, “lost in the woods”.
4.6	*Poor Risk Management*: Countering the wrong risks.
4.7	*Choosing the Wrong Development Strategy*: e.g. waterfall, prototyping, etc.
5.	**Scope**
5.1	*Unclear/Misunderstood Scope/Objectives*. It is impossible to pin down the real scope or objectives due to differences or fuzziness in the user community.
5.2	*Changing Scope/Objectives*: Business changes or reorganizes part way through the project.
5.3	Scope Creep: Not thoroughly defining the scope of the new system and the requirements before starting, consequently not understanding the true work effort, skill sets and technology required to complete the project.
5.4	*Project Not Based on Sound Business Case*: Users and developers ignore business requirements, develop system for sake of technology.
5.5	*Number of organizational units involved*: increased number of lines of communication and conflict potential expands the scope of the system.
6.	**Requirements**
6.1	*Lack of Frozen Requirements*. Because the needs of the users change, the requirements change. Consequently the system will never be moved into production because none of the requirements are ever completed. Alternatively, freezing a subset of the functionality and delivering allows for the completion of the system and update releases as required.
6.2	*Misunderstanding the Requirements*. Not thoroughly defining the requirements of the new system before starting, consequently not understanding the true work effort, skill sets and technology required to complete the project.
6.3	*New and/or Unfamiliar Subject Matter for Both Users and Developers:* Lack of domain knowledge leads to poor requirements definition.
6.4	*Inadequate documentation of user requirements*
7.	**Funding**
7.1	*Under Funding of Development*: Setting the budget for a development effort before the scope and requirements are defined or without regard to them (i.e., picking a number out of the air).

(continues)

TABLE 31 (continues)

7.2	*Under Funding of Maintenance*: Support for products in the maintenance phase. If the customer is unprepared or does not budget for this, the project can be judged a failure even if successful in all other aspects.
7.3	*Bad Estimation*: Lack of effective tools or structured techniques to properly estimate scope of work. Unrealistic cost estimates cause illogical or sub-optimal planning, strategy, and decisions.
7.4	*"All or Nothing":* Requires budgeting entire project at the outset, leading to under funding in later years of project.
7.5	*No investments to IT*: Investors are cautious to put money on software development.
7.6	*Huge capital requirements*: There is no efficient leasing program in the country (hardware is expensive)
7.7	*Poverty of software companies*: employing staff is very expensive, and the wages serve as catalyst for them to be committed
8.	**Scheduling**
8.1	*Artificial Deadlines*. Presence of unrealistic deadlines or functionality expectations in given time period. - 'crash projects' in which test time or training time is reduced – using something other than work effort required to determine when the new system should move into production.
8.2	*"Preemption" of Project by Higher Priority Project*: Management unable to resolve conflicting schedule demands.
8.3	*Lack of experience of the user management*: timing demands are unrealistic
9.	**Development Process**
9.1	*Lack of Effective Development Process/Methodology:* Leading to quality problems - Documentation, Software and Testing—poor estimating -- insufficient time for up-front work, e.g., design—little flexibility for change—insufficient testing.
9.2	*Trying New Development Method/Technology During Important Project*
9.3	*Lack of quality standards*
9.4	*Lack of computing literature*: no literature for development process or techniques
10.	**Personnel**
10.1	*Lack of Required Knowledge/Skills in the Project Personnel*: e.g., technology, business knowledge and experience.
10.2	*Lack of "People Skills" in Project Leadership*: PM tries to "manage" schedules, technology, requirements, etc., ignoring that management is dealing with people on the team.
10.3	*Poor Team Relationships*: Strains existing in the team due to such things as burnout or conflicting egos and attitudes.
11	**Staffing**
11.1	*Insufficient/Inappropriate Staffing*: Not enough people or people with wrong skills/insufficient skills assigned to project, regardless of availability.
11.2	*Staffing Volatility*: At some point in the project, losing the key project manager, analysts or technicians (especially in new technology).
11.3	*Excessive Use of Outside Consultants*: Can lead to a conflict of interest, e.g., billable hours vs. budget, or resulting in the internal staff not having significant involvement
11.4	*Lack of Available Skilled Personnel*: People with the right skills are not available when you need them.
12.	**Technology**
12.1	*Introduction of New Technology:* Using new, or 'bleeding edge', technology that has not been used successfully at other companies, or major technological shift occurs during the project.
12.2	*Stability of Technical Architecture* – Has to be done before comparable applications.
12.3	*Inappropriate technology*: Trying to achieve a particular task/project without the appropriate tools

(continues)

TABLE 31 (continues)

13.	**External Dependencies**
13.1	*External Dependencies Not Met*: The project's consultants or vendors do not deliver, go out of business, or are unclear as to their roles and responsibilities.
13.2	*Multi-Vendor Projects Complicate Dependencies*: Integration of packages from multiple vendors hampered by incompatibilities and lack of cooperation between vendors.
13.3	*Lack of Control Over Consultants, Vendors, and Sub-contractors*: Schedule or quality problems beyond control of project manager. No legal recourse due to poor contract specification.
13.4	Importation of foreign packages: craze for foreign packages with cheep price do not allow the growth of indienous developers
14.	**Planning**
14.1	*No Planning or Inadequate Planning*: Attitude that planning is unimportant or impractical.

Note: Items are grouped by category. Shaded items represent risk factors named in Nigeria.

APPENDIX 4

TABLE 32 Comparison of risk factors to Schmidt et al. (2001) study.

1.	**Corporate Environment**	**Observed by Barki and Boehm**	**Observed in Nigeria**
1.1	*A climate of change in the business and organizational environment that creates instability in the project.*	No	1.1
1.2	*Mismatch between company culture and required business process changes needed for new system*. A mismatch between the corporate culture and the changes required by the new system.	No	No
1.3	*Projects That Are Intended to Fail*: Projects started for political reasons which carry no clear business value but serve to divert the organization's focus from actual needed change. Such projects are under-funded, not supported and are not intended to succeed. Projects have no business value and are used as diversionary tactics to avoid facing the real change needs.	No	No
1.4	*Unstable Corporate Environment:* Competitive pressures radically alter user requirements, sometimes making the entire project obsolete.	No	1.2
1.5	*Change in Ownership or Senior Management:* New owners and/or managers set new business direction that causes mismatch between corporate needs and project objectives.	No	No
2.	**Sponsorship/Ownership**		
2.1	*Lack of Top Management Commitment to the Project*. This includes oversight by executives and visibility of their commitment, committing required resources, changing policies as needed.	Yes	2.1 *
2.2	*Lack of client responsibility, ownership, and buy-in of the project and its delivered system(s).*	Yes	2.2
2.3	*Failure to gain user commitment*: Laying blame for "lack of client responsibility" on the project leader rather than on the users.	No	2.3 *
2.4	*Conflict Between User Departments*: Serious differences in project goals, deliverables, design, etc., calls into question concept of shared ownership.	Yes	No
2.5	*Failure to get project plan approval from all parties*	No	No

(continues)

TABLE 32 (continues)

3.	**Relationship Management**		
3.1	*Failure to Manage End User Expectations*: Expectations determine the actual success or failure of a project. Expectations mismatched with deliverable - too high or too low - cause problems. Expectations must be correctly identified and constantly reinforced in order to avoid failure.	No	3.1
3.2	*Lack of Adequate User Involvement*: Functional users must actively participate in the project team and commit to their deliverables and responsibilities. User time must be dedicated to the goals of the project.	No	3.2
3.3	*Lack of Cooperation from Users*: Users refuse to provide requirements and/or refuse to do acceptance testing.	Yes	3.3
3.4	*Failure to Identify All Stakeholders*: Tunnel vision leads project management to ignore some key stakeholders in the project, affecting requirements definition, implementation, etc.	No	No
3.5	*Growing Sophistication of Users Leads to Higher Expectations*: Users are more knowledgeable, have seen sophisticated applications, apply previous observations to existing project.	No	3.4
3.6	*Managing Multiple Relationships with Stakeholders*: Some "clients" are also "partners" in producing deliverables in other projects. Leads to confusion of roles and responsibilities.	No	No
3.7	*Lack of appropriate experience of the user representatives*: Users assigned who lack necessary knowledge of the application or the organization	Yes	3.5
4.	**Project Management**		
4.1	*Not Managing Change Properly:* Each project needs a process to manage change so that scope and budget are controlled. Scope creep is a function of ineffective change management and of not clearly identifying what equals success.	No	4.1
4.2	*Lack of Effective Project Management Skills*: Project teams are formed and the project manager does not have the power or skills to succeed. Project administration must be properly addressed.	No	4.2
4.3	*Lack of Effective Project Management Methodology*: The team employs no change control, no project planning or other necessary skills or processes.	No	4.3

(continues)

TABLE 32 (continues)

4.4	*Improper Definition of Roles and Responsibilities*: Members of the project team and the organization are unclear as to their roles and responsibilities. This includes outsourcers and consultants.	Yes	4.4
4.5	*Poor or Non-Existent Control*: No sign-offs, no project tracking methodology, unaware of overall project status, "lost in the woods".	No	No
4.6	*Poor Risk Management*: Countering the wrong risks.	No	4.5
4.7	*Choosing the Wrong Development Strategy*: e.g. waterfall, prototyping, etc.	No	No, but was chosen later *
5.	**Scope**		
5.1	*Unclear/Misunderstood Scope/Objectives*. It is impossible to pin down the real scope or objectives due to differences or fuzziness in the user community.	No	5.1 *
5.2	*Changing Scope/Objectives*: Business changes or reorganizes part way through the project.	No	No, but was chosen later *
5.3	Scope Creep: Not thoroughly defining the scope of the new system and the requirements before starting, consequently not understanding the true work effort, skill sets and technology required to complete the project.	Yes	5.2
5.4	*Project Not Based on Sound Business Case*: Users and developers ignore business requirements, develop system for sake of technology.	Yes	No
5.5	*Number of organizational units involved*: increased number of lines of communication and conflict potential expands the scope of the system.	Yes	No
6.	**Requirements**		
6.1	*Lack of Frozen Requirements*. Because the needs of the users change, the requirements change. Consequently the system will never be moved into production because none of the requirements are ever completed. Alternatively, freezing a subset of the functionality and delivering allows for the completion of the system and update releases as required.	Yes	6.1
6.2	*Misunderstanding the Requirements*. Not thoroughly defining the requirements of the new system before starting, consequently not understanding the true work effort, skill sets and technology required to complete the project.	Yes	6.2 *

(continues)

TABLE 32 (continues)

6.3	*New and/or Unfamiliar Subject Matter for Both Users and Developers:* Lack of domain knowledge leads to poor requirements definition.	Yes	6.3
7.	**Funding**		
7.1	*Under Funding of Development*: Setting the budget for a development effort before the scope and requirements are defined or without regard to them (i.e., picking a number out of the air).	Yes	7.1 *
7.2	*Under Funding of Maintenance*: Support for products in the maintenance phase. If the customer is unprepared or does not budget for this, the project can be judged a failure even if successful in all other aspects.	Yes	No
7.3	*Bad Estimation*: Lack of effective tools or structured techniques to properly estimate scope of work. Unrealistic cost estimates cause illogical or sub-optimal planning, strategy, and decisions.	Yes	7.2
7.4	*"All or Nothing":* Requires budgeting entire project at the outset, leading to under funding in later years of project.	Yes	No
8.	**Scheduling**		
8.1	*Artificial Deadlines*. Presence of unrealistic deadlines or functionality expectations in given time period. - 'crash projects' in which test time or training time is reduced - using something other than work effort required to determine when the new system should move into production.	Yes	8.1 *
8.2	*"Preemption" of Project by Higher Priority Project*: Management unable to resolve conflicting schedule demands.	Yes	No
9.	**Development Process**		
9.1	*Lack of Effective Development Process/Methodology:* Leading to quality problems - Documentation, Software and Testing—poor estimating -- insufficient time for up-front work, e.g., design—little flexibility for change—insufficient testing.	No	9.1 *
9.2	*Trying New Development Method/Technology During Important Project*	No	No, but was chosen later *
10.	**Personnel**		
10.1	*Lack of Required Knowledge/Skills in the Project Personnel*: e.g., technology, business knowledge and experience.	Yes	10.1 *
10.2	*Lack of "People Skills" in Project Leadership*: PM tries to "manage" schedules, technology, requirements, etc., ignoring that management is dealing with people on the team.	No	10.2 *

(continues)

TABLE 32 (continues)

10.3	*Poor Team Relationships*: Strains existing in the team due to such things as burnout or conflicting egos and attitudes.	Yes	10.3
11	**Staffing**		
11.1	*Insufficient/Inappropriate Staffing*: Not enough people or people with wrong skills/insufficient skills assigned to project, regardless of availability.	Yes	11.1
11.2	*Staffing Volatility*: At some point in the project, losing the key project manager, analysts or technicians (especially in new technology).	Yes	11.2
11.3	*Excessive Use of Outside Consultants*: Can lead to a conflict of interest, e.g., billable hours vs. budget, or resulting in the internal staff not having significant involvement	No	No
11.4	*Lack of Available Skilled Personnel*: People with the right skills are not available when you need them.	No	11.3 *
12.	**Technology**		
12.1	*Introduction of New Technology:* Using new, or 'bleeding edge', technology that has not been used successfully at other companies, or major technological shift occurs during the project.	Yes	12.1
12.2	*Stability of Technical Architecture* – Has to be done before comparable applications.	No	No
13.	**External Dependencies**		
13.1	*External Dependencies Not Met*: The project's consultants or vendors do not deliver, go out of business, or are unclear as to their roles and responsibilities.	Yes	No
13.2	*Multi-Vendor Projects Complicate Dependencies*: Integration of packages from multiple vendors hampered by incompatibilities and lack of cooperation between vendors.	Yes	No
13.3	*Lack of Control Over Consultants, Vendors, and Sub-contractors*: Schedule or quality problems beyond control of project manager. No legal recourse due to poor contract specification.	No	No
14.	**Planning**		
14.1	*No Planning or Inadequate Planning*: Attitude that planning is unimportant or impractical.	No	14.1

Note: The comparison includes also Boehm's and Barki's factors (i.e., Boehm 1989, Barki, et al. 1993). Asterisks in Nigerian factors indicate risks in top 19 list, chosen for ranking.

APPENDIX 5

TABLE 33 Sustainability factors (as indicated by IT experts in Nigeria)

1. The level of demand of technology

- Fore sight: dynamic societal needs / requirements requires that the design and system be proactive, should have a vision and incorporate the anticipated future use to their designers.
- Purposeful: Software applications must satisfy a definite need in the organisation. An application are bought with no clear need for it, it would not be sustained
- Sharper focus: Applications must address smaller and more specified needs.
- Study of customer requirement.
- Efficiency in data handling: system must be fast enough to cope with increased business volume.
- Having an IT vision.

2. The appropriateness of the technology to the application environment

Cost-effectiveness

- Feedback: Users/usage opinion must be sought constantly.
- Reliability: The software must solve the problem at hands and the output must be reliable.
- A good customer feedback media
- Meet the requirements of the market: does the required job appropriately.

Affordability

- Adequate funding.
- Good capital base.
- Adherence to operational rules.
- Side effects: The software must have little or no negative side effects on the user's current system.

Suitability

- User friendly: User friendly applications are suitable to new entrants in the IT dept.
- Easy to use: software applications should be easy to understand, install and deploy to a wide range of personnel.
- Usability: It must be user friendly. Interface must not be ugly.
- User friendly: ability of users to relate easily with the application.
- Simplicity: systems that are simple in design and implementation have proven to used for a long-term basis.
- Ease of use: users must be able to use the software with minimal education and supervision.
- Ease of generating reports – MIS.
- Interface.
- Scaleable: Software applications should be used on any machine and most operating system.
- Platform independence: Applications should not be restricted to a particular HW or development platform.
- Open interface: Applications must allow means for other systems to exchange data with them.
- Dynamic development environment: the platform of design should be dynamic with continuously emerging technology. This reduces risk of obsolescence.
- Portability: ability to migrate your software from one platform to another.
- Flexible: Sustainable system must not be rigid. It must exist for all kinds of business practices.
- Flexibility: adaptability of software to changing modes of operation of users.
- Flexibility: 'closed-in' systems raise the cost of maintenance.
- Generational compatibility: how easily is application migrated for one platform.
- Interoperability.

(continues)

TABLE 33 (continues)

- Multi-user: software applications should be centralized with everyone accessing it from one point.
- Adaptability to changing in policy in transparent manner (changes can be handled by data, not by changing softa).
- Maintenance: Adequate maintenance of infrastructure e.g. computers, network, etc.
- Maintainability: Sustainable application must be such that is easily maintained or customized to users need.
- Cost of maintenance: The cost of acquiring and maintaining the software must be as minimal as possible.
- Availability and support: Post implementation support capabilities are necessary for long-term usability. Lack of support has lead to good application being abandoned.
- Improvement: continuous upgrade of software, providing addition facilities.
- Ease of future modification.
- Parameterisation of the applications: enables modifications without programming.
- Availability of documentation.
- Documentation: A good documentation of the technical aspect as well as users manual must be done.
- Documentation: insufficient technical documentation can seriously affect support capabilities.
- Proper documentation of programs: proper documentation of program allows for continuity, ease of maintenance even when the original developers are no being available.
- Reusability of code & design.
- Consistent design and programming style so that programmers can easily understand and maintain the system.
- Usability of codes, reuse of frameworks, object-oriented so any problem can be solved by others
- Error free: The application should have minimal bugs/errors.
- Effective & throughout testing: this exposes in your system the lacks and enables rebuild before launching out.
- Testing: Application must be well tested using users raw data.
- Ensuring software products go through necessary quality checks.
- Up-to-date upgrade.
- Date formatting: e.g. year 2000 problems.
- Reliability: security and integrity of data must be assured.
- Reliability and durability of application.
- Stability in production use : reliable system.

<u>3. The availability of local technological capacity to sustain its beneficial use</u>

In-house department

- Ease of support by in-house personnel.
- Available support/maintenance: Availability of human resources to support / maintain the product.
- How big IT-department in client side .
- Having technical team inside (in client side), who can support (in consultation).

Organizational commitment

- Commitment.
- Commitment of user organisation's top-mgt to IT.
- Support from top management.
- Committed and result-oriented users and management.
- User involvement at design phase.
- Customer relationship.
- Customers: Involvement of users (consultant) in the area of system being developed must be included.
- More user involvement in projects.

(continues)

TABLE 33 (continues)

- Staff loyalty.
- Commitment from all involved is a serious success factor. The developers , the users and those who pays for the project agree to work as partners.

User training

- Inadequate user training: reduces optimal use of software, prolongs implementation support.
- Proper training of users: users will be self-supporting.
- Technical expertise should be encouraged.
- Availability of technical support: users will be self-supporting.

4. Development process

- Selection of proper / appropriate development tool.
- Programming tool: must be flexible and should allow easy debugging.
- Provision of necessary tools.
- Proper selection of development tools: tools must be carefully selected to ensure availability of upgrades to suit changing IT environment.
- Inappropriate development tool: programming tools without upgrade capability or developer backup.
- Deployment of a good design methodology: Selecting and adopting sustainable standard development method ensures sustainable applications.
- Experience: Sustainable applications are built on experience and previous applications. The planning and monitoring of implementation should therefore be left to experienced hands. Trainees can handle coding.
- Programmers: must be well co-ordinated..
- Good team work.
- Adequate planning.
- Thorough systems design and planning.
- Parametrization.
- Modular design.
- Good design background: If a product is to be used for a long period, it must be on a good design platform.
- Through analysis and specification design: This ensures full understanding of what to do and how best to present your solution. Useful specifications are built during this phase.

1 ROPPONEN, JANNE, Software risk management - foundations, principles and empirical findings. 273 p. Yhteenveto 1 p. 1999.
2 KUZMIN, DMITRI, Numerical simulation of reactive bubbly flows. 110 p. Yhteenveto 1 p. 1999.
3 KARSTEN, HELENA, Weaving tapestry: collaborative information technology and organisational change. 266 p. Yhteenveto 3 p. 2000.
4 KOSKINEN, JUSSI, Automated transient hypertext support for software maintenance. 98 p. (250 p.) Yhteenveto 1 p. 2000.
5 RISTANIEMI, TAPANI, Synchronization and blind signal processing in CDMA systems. - Synkronointi ja sokea signaalinkäsittely CDMA järjestelmässä. 112 p. Yhteenveto 1 p. 2000.
6 LAITINEN, MIKA, Mathematical modelling of conductive-radiative heat transfer. 20 p. (108 p.) Yhteenveto 1 p. 2000.
7 KOSKINEN, MINNA, Process metamodelling. Conceptual foundations and application. 213 p. Yhteenveto 1 p. 2000.
8 SMOLIANSKI, ANTON, Numerical modeling of two-fluid interfacial flows. 109 p. Yhteenveto 1 p. 2001.
9 NAHAR, NAZMUN, Information technology supported technology transfer process. A multi-site case study of high-tech enterprises. 377 p. Yhteenveto 3 p. 2001.
10 FOMIN, VLADISLAV V., The process of standard making. The case of cellular mobile telephony. - Standardin kehittämisen prosessi. Tapaustutkimus solukkoverkkoon perustuvasta matkapuhelintekniikasta. 107 p. (208 p.) Yhteenveto 1 p. 2001.
11 PÄIVÄRINTA, TERO, A genre-based approach to developing electronic document management in the organization. 190 p. Yhteenveto 1 p. 2001.
12 HÄKKINEN, ERKKI, Design, implementation and evaluation of neural data analysis environment. 229 p. Yhteenveto 1 p. 2001.
13 HIRVONEN, KULLERVO, Towards Better Employment Using Adaptive Control of Labour Costs of an Enterprise. 118 p. Yhteenveto 4 p. 2001.
14 MAJAVA, KIRSI, Optimization-based techniques for image restoration. 27 p. (142 p.) Yhteenveto 1 p. 2001.
15 SAARINEN, KARI, Near infra-red measurement based control system for thermo-mechanical refiners. 84 p. (186 p.) Yhteenveto 1 p. 2001.
16 FORSELL, MARKO, Improving Component Reuse in Software Development. 169 p. Yhteenveto 1 p. 2002.
17 VIRTANEN, PAULI, Neuro-fuzzy expert systems in financial and control engineering. 245 p. Yhteenveto 1 p. 2002.
18 KOVALAINEN, MIKKO, Computer mediated organizational memory for process control. Moving CSCW research from an idea to a product. 57 p. (146 p.) Yhteenveto 4 p. 2002.
19 HÄMÄLÄINEN, TIMO, Broadband network quality of service and pricing. 140 p. Yhteenveto 1 p. 2002.
20 MARTIKAINEN, JANNE, Efficient solvers for discretized elliptic vector-valued problems. 25 p. (109 p.) Yhteenveto 1 p. 2002.
21 MURSU, ANJA, Information systems development in developing countries. Risk management and sustainability analysis in Nigerian software companies. 296 p. Yhteenveto 3 p. 2002.