Dynamic Knowledge Integration

- A field study of an Information Systems Development Project

LINNÉA WAHLSTEDT

Linköping University

Linköping Studies in Arts and Science No. 603
Linköping University, Sweden
Department of Management and Engineering
Linköping 2014

Linköping Studies in Arts and Science, No. 603

At the Faculty of Arts and Science at Linköping University, research and doctoral studies are carried out within broad problem areas. Research is organized in interdisciplinary research environments and doctoral studies mainly in graduate schools. Jointly, they publish the series Linköping Studies in Arts and Science. This thesis comes from Business Administration at the Department of Management and Engineering

Distributed by:
Department of Management and Engineering
Linköping University
SE-581 83 Linköping
SWEDEN

Linnéa Wahlstedt
Dynamic Knowledge Integration
A field study of an Information Systems Development Project

Upplaga 1:1
ISBN 978-91-7519-438-7
ISSN 0282-9800

Printed in Sweden by LiU-Tryck, Linköping, Sweden, 2014

Abstract

Current research on knowledge integration offers valuable structural analyses of factors that influence knowledge integration, performance outcomes, and knowledge integration mechanisms. Less attention has been paid to how knowledge integration is carried out over time in cross-functional development projects. This thesis is based on a year-long field study of an Information Systems Development Project. The study shows how the knowledge integration process was repeatedly interrupted by different problems that could not be resolved by merely relying on integration mechanisms that were imposed by the top management. Instead, a bottom-up dynamic evolved where the project members and participating project managers managed to reestablish coordination and knowledge integration through the invention of different 'collective heuristics'. A novel model of Dynamic Knowledge Integration is presented which claims that knowledge integration contains two interplaying processes; one consisting of different knowledge integration mechanisms and activities, and one consisting of the collective heuristics that were invented and employed when unexpected problems emerged. In general, this research argues that knowledge integration can be understood as a dynamic process, of which both knowledge integration mechanisms and collective heuristics constitute core elements.

To my beloved brothers

Alexander and David

I miss you

Acknowledgements

I can hardly believe it (not my dear family either) but now it is time for me to write the last part of this dissertation – the acknowledgements. I am so proud and grateful that I have eventually come to this moment. Even though the writing of the thesis has involved hundreds of hours of working alone in front of the computer, this is not a one-person accomplishment. The thesis is rather a product of a close interaction with many different people who deserve my deepest gratitude.

First of all, I would like to thank my advisor, Lars Lindkvist. Your impressing ability to pick up interesting things in messy material and unclear drafts has taken this research far beyond what I thought was possible. Your brilliant comments, painful truths and pedagogical reading have also resulted in a feeling of trust and confidence. We have had interesting conversations on all aspects of life and also a lot of fun, especially on our research travels around the world. I would not have coped with all this work during the years if our relationship had only been formal or impersonal. I hope our collaboration will continue in the future so that we can complain about the food on the plane again and hunt for a crossword and a 'French Nougat' at the train station once more!

I am also particularly grateful for Jonas Söderlund's excellent remarks on early as well as late versions of the thesis. When I believed that I had thought of 'everything' you showed that important questions remained to be answered which pushed my mental limits and improved the outcome considerably.

Christine Räisänen has also provided careful observations, invaluable questions and useful recommendations. When I look back on the material that you were given a couple of years ago I realize that it must have been a tough task to generate such an insightful commentary as you succeeded with. I would also like to thank Fredrik Tell who has read the manuscript and encouraged my thinking, especially at an internal seminar in 2010 when I needed it the most.

A very special thank you goes to Jon who welcomed me to PPM and made the empirical study at PPM possible. The line managers, various line members, board members and the project members, especially Tom-Erik, Jenny, Stefan, Johan, Bosse, Jahan, Elle, Karin, Eva, Carina, Pia, Ulrik, and Olle, have patiently answered my questions and let me join their meetings and 'hang around' for a year. I have learned so much about the 'real life' from you.

FAS, which now is called Forte (Swedish Research Council for Health, Working Life and Welfare), have contributed to this research by being the main external financial sponsor. I am very grateful for that.

My colleagues at the Department for Management and Engineering at Linköping University and in particular the fellows of my research group have been of great help during the whole process. Whatever I have struggled with and irrespective of topic there has always been someone to share problems and talk to. Thank you Karin, Cecilia, Elisabeth, Ramsin, Christofer, Marie, and all the rest of you, for being interested, helpful, and positive.

Especially Marie has meant much to me during these years, not only for being a tremendous and thoughtful listener with an incredible talent to find solutions to different tricky problems, but also for being a really good friend who truly understands what it is like to be a PhD

student in my situation. Peculiar situations and joyful moments - memories abound from our joint trips and adventures. However, the best thing I learned from you (beside the fact that food can taste quite well even without vegetables) is the relationship between goals, expectations, and disappointments. I am always on my way somewhere with high expectations and a specific goal in mind. This means that I often run the risk of being terribly disappointed if something happens on the way that hinders me from reaching the destination. Fewer expectations involve fewer disappointments and more room for unplanned discovery and appreciation of the means and not merely the goals. This means more and more to me in different aspects of life.

My dear friends Jenny, Linda, Linda S, Andrea, and Helena, are you still interested in the dinners, lunches, skiing trips, travels, weekends, and family get-togethers that you were talking about? Since I have suddenly got some more time I will say yes now to all your ideas! I really look forward to seeing you a bit more often again. Thank you for your support and comfort and for just being there.

I also want to thank WS and all the friendly people I have met there. You have certainly had a great influence on this work since you have offered a nice, warm and open work context where I could discipline myself, organize the work, and get inspiration and motivation to finally complete this project.

Life is not always kind to people. I wish everyone had someone like you, Lisbet. Thank you for all your loving care during these years. Cecilia, you are like an extra family member for us. We have known each other for a long time now and share many experiences of life. Thank you for all your help with Amelie the past 5 years. My 'parents in law' Gun and Lasse, I am so grateful for everything you have done for us. The kids love to be with you, Gun - their warm, tender and playful grandma - which has been of great importance to me since I have had to ask for your help almost every week the last year. Lasse, our skilled furniture maker and positive grandpa, without your work on our house we would still have been living as in a storage room, and for the third summer been eating dinners right on the empty balcony deck.

Mum, Dad, and Lelle, you are the tireless cheerleaders every doctoral student needs. Thank you for constantly expressing your faith in me. Thank you for taking care of the kids every week. Thank you for being ready to help with anything 24/7. Thank you for driving me home in the middle of the night. Thank you for late dinners, fancy lunches and material and financial support. I could extend this list with dozens of things. The essence is that without all your help I would still be surrounded by chaotic texts and be lost in unclear thoughts with tears in my eyes. This work would never have been finished without you. Thank you for your endless love.

My last innermost thank you goes to my family at home. My treasures, Amelie and Noel, I promised you to complete the book before Christmas, actually before the Lucia celebration in the middle of December. I cannot let you down; I must meet this deadline. This is actually how you have contributed to the completion of the thesis – by constantly asking when I am finished and when I will come back home from work. A couple of days ago when I had been working days and nights for some time and barely slept at all, Amelie asked me, while looking out from the window "Mama, now everyone has put up Christmas lights, why do not we have any Christmas lights?" My sweetheart, Mum's book is finally finished and I will put up as many lights and candles and prepare for Christmas as I have never done before! But most of all, I will embrace the two of you, Amelie and Noel, and hold you tight for a really long time. Knowing that I soon will be able to spend more time with you has been the absolute strongest driving-force behind the completion of this thesis. I love you so much.

Peter, we both have felt that the dissertation has intruded upon our possibilities to live an ordinary family life. Every weekend, every holiday, and almost every decision have been influenced by the dissertation. Thank you for tolerating this abnormal way of living and thank you for your optimistic imagining and constant belief in the time that will come afterwards. I love listening to your wise words, which always leaves a sense of hope and happiness in my body. Thank you for sharing this from the first until the last minute. You are simply fantastic! We will find a new way of living now. We will spend time together in the weekends, make up plans for joint holidays and have time to just live and enjoy whatever life offers. A new era is waiting for us. These are beautiful thoughts. However, to start with, you can leave the stomach sick kids with me now and calmly go back to work again. Leave the beds unmade and just leave the dishes. Come back home when you get hungry. I will cook for you because now, I'm finished.

Linnéa

Stockholm December 11, 2013

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF QUOTED PROJECT PARTICIPANTS

NAME	SUBPROJECT	ROLE	LINE DEPARTMENT/ CONSULTANT
Ben	IT	IT Programming Leader, K2 Programming Leader	IT
Adam	IT	ELWIS Developer	IT
John	IT	IT Project Manager Sep 2002- May 2003. Design Expert May 2003- Nov 2004	IT
Simon	IT	Designer 2002-2003. Design Leader May 2003-Nov 2004	Consultant
Olle	IT Project Office	Quality Controller/Configuration Manager	IT
Jack	IT	ELWIS Programming Leader Jan 2004-Nov 2004	IT
Martina	IT	IT Subproject Manager May 2003- Jan 2004	IT
Carl	IT	ELWIS Developer, Database Expert	IT
Thomas	IT	Head Project Manager May 2003-Nov 2004. IT Subproject Manager Jan 2004-Nov 2004	Consultant
Anonymous Developer 1-3	IT	Developer, communication channels.	IT
Anonymous ELWIS Developer	IT	ELWIS Developer	IT
Elisabeth	Requirement. Operations.	Project Manager Requirement Project. Sep 2002-May 2003. Requirement Coordinator May 2003-Nov 2004	Insurance. Consultant
Kristina	Requirement. Operations.	Requirement Project Member. Operations Subproject Manager May 2003-Nov 2004	Insurance

Jessica	Operations Implementation and Data Conversion.	Operations Subproject member May 2003- Nov 2004. Subproject Manager Implementation and Data Conversion May 2003-Nov 2004.	IT
Caroline	Requirement. Operations.	Requirement and Operations Subproject Member.	Communications department. Client Services.
Louise	Requirement. Operations.	Requirement and Operations Subproject Member.	Communications department. Pension unit.
Elsie	Requirement. Operations.	Requirement and Operations Subproject Member.	Communications department. Pension Unit Manager.
Workshop Participants 1-5	Requirement. Operations.	Requirement and Operations Subproject Members.	Insurance, Law unit, Communications, Planning and Administration.
Jesper	Test	Test Expert	IT
Urban	Test. Requirement. Operations.	Requirement and Operations Subproject Member. Acceptance tester.	Communications department. Client Services.

CHAPTER I

I. INTRODUCTION

A decade ago, a certain group of people in Sweden with different skills and knowledge was assigned to establish a new pension system and develop information systems and computer programs that were to be associated with this pension system. This proved to be a challenging task, which lasted for several years. Besides the task's inherent complexity, the work also gave rise to many difficult situations and problems that were related to collaboration, communication, and the integration of knowledge. The first years of the task were filled with conflict, confusion, and bad results. The people who were involved in the project did not seem able to use their various capabilities and expertise to create useful concepts and products. In the middle of the project, even more problems arose in connection with differences in work habits, preferences, and interests. Even though a broad set of skills and knowledge was needed to solve this complex task and accomplish a creative solution, the knowledge-related diversity that was manifest in the group made it also difficult for the participants to collaborate and find well-functioning forms of interaction. This, in turn, hindered progress and the creation of a coherent solution. Many changes in the organization were made so as to improve interaction and the integration of expertise. In the last part of the process, the already time-pressured schedule was cut short, which intensified the pressure on the workers who could no longer continue with the task according to the plan, or in accordance with the pre-determined work method. This put great demands on the collective adaption of the work model and the creation of new forms of collaboration – which were needed to speed up the integration of the workers' different contributions.

The thesis focuses on how different problems related to knowledge integration were solved in one of the development projects that were formed to establish the new pension system and its related IT-systems. Under certain circumstances, it seemed appropriate to make changes in the organization in order to progress which promoted the expression of differences and allowed for more diversity. These changes were made in order to accomplish a creative and complex solution. In other situations, the opposite action appeared to be more suitable; that is to say, possibilities for expressing different perspectives and conducting divergent actions were restricted in order to move forward more quickly and to produce a coherent outcome. Understanding how to manage the contrasting demands, on the one hand, of maximizing and using people's diverging skills and knowledge to achieve creative and complex products and, on the other hand, of reducing the space for using knowledge diversity to move forward quickly and to produce integrated and coherent products is focused on in this research. The challenge of promoting and utilizing the participants' differences in knowledge and skills whilst at the same time ensuring that the development process proceeds in a timely manner, in accordance with plans and specifications, is thus of particular interest to this study.

This thesis takes its starting point in the area of *knowledge integration*. Knowledge integration is viewed here as a matter of combining different knowledge bases in new ways so as to create new concepts and products. This view means that minimal effort is expended on knowledge sharing and cross-learning activities among diverse people. The integration of different knowledge bases, instead, should be accomplished through the use of different knowledge integration mechanisms, such as sequencing, routines, rules and directives, and group problem-solving and decision-making (Grant 1996a). The study aims to contribute to the understanding of knowledge integration in several ways. First, it examines the practical use of different knowledge integration mechanisms in an on-going business. This is done so as to understand how key people in the development process, who could be both project managers and project members, actually combine their diverse knowledge and engage in knowledge integration. Second, the study seeks to understand how knowledge integration mechanisms are used in different phases of the project and how different knowledge integration mechanisms may change over time. This is of interest because different development stages and conditions may involve different challenges that require different combinations of knowledge integration means. Third, one could imagine that the project members also invent new means, techniques, tools or strategies to solve unexpected problems and enhance knowledge integration and project progress, which this study also seeks to identify. Fourth, the study enquires into how knowledge integration mechanisms can be used to manage knowledge-related diversity in different project phases and in different problematic situations. This is summarized in two research questions:

1. What knowledge integration mechanisms and other collective means to enable knowledge integration and project progress can be identified throughout the project process?

2. How are the identified knowledge integration mechanisms and the other knowledge integration enablers used to extend or limit the space used for expressing knowledge differences in different project phases in the face of the approaching deadline?

In light of the above, the purpose of the study is to understand how project groups manage the process of knowledge integration in a complex development context which is characterized by expertise diversity and a limited amount of time with which to complete the task on hand. It seeks to provide understanding of how project members use various knowledge integration mechanisms over time to integrate knowledge and strike a balance between making the most of project members' heterogeneity of expertise and at the same time meeting the demand for time-efficient coordination and swift progress. This will involve the identification of different knowledge integration mechanisms for modulating the space that is available for expressing individual differences. It will also include a description of how specific knowledge integration mechanisms may vary over time. In addition, the study presents an examination of how project members themselves invent and use a variety of helpful measures to facilitate progression at different stages of the development process. The general ambition of the study is to contribute to current research on knowledge integration by adding a process understanding of the utilization and creation of knowledge integration mechanisms in complex development settings.

The thesis consists of eight chapters. After this introductory chapter where the empirical study, the problem of interest, the conceptual starting point, and the study's destination are shortly introduced, the reader will find a theoretical discussion in Chapter 2. Here the theoretical knowledge domain to which this research belongs and contributes to is presented and discussed. Chapter 2 offers insight into the field so that we will know what to look for and pay extra attention to in the empirical case. Before proceeding to the empirical section, the reader will find a chapter that presents the research method and takes the reader on a tour 'behind the scenes' as it were. The research method consists of a detailed ethnography, which is explained and described in Chapter 3. The following chapter, Chapter 4, is the most comprehensive section of the thesis. It comprises of an ethnographic description of an information systems development (ISD) project. I entered the field when the project had already commenced with its second year, thus the project's first year was studied in retrospect (on site) through conducting interviews and examining documents. In Chapter 5, I summarize and interpret the project’s events, problems, and solutions related to knowledge integration and diversity management. The following chapter, Chapter 6, is a synthesis of the interpretation that is presented in Chapter 5. Here, the level of abstraction moves up, and in Chapter 6 I suggest and discuss a new concept. In the penultimate chapter, Chapter 7, a discussion that relates and compares the new concept with previous research can be found. The final chapter of this thesis, Chapter 8, contains the study’s conclusions and contributions and includes answers to the research questions, as well as suggestions for future research.

CHAPTER II

II. KNOWLEDGE INTEGRATION IN THEORY

Conceptual Point of Departure

1. "Terra incognita"

Integrating different experts' knowledge bases is a critical activity in technology-based innovation (Berggren et al. 2011). In order to solve complex problems and carry out multifaceted tasks, such as information systems development where a specific project goal is identified from a broad business concept, where system requirements are formulated, and useful software solutions developed, various people's skills, experiences, and contributions must be used and combined. Tiwana and McLean (2005) studied knowledge integration in information system development (ISD) and stated that a team's collective creativity and ability to deliver useful, coherent and creative solutions depends on the capability to integrate the different project members' expertise. "When integration of team members' expertise at the project level is poor, the team might build its *ignorance* – unstated requirements, evolving user needs, unrecognized constraints, and incomplete understanding of the problem domain – into the design of the system." Tiwana and McLean (2005 p. 19)

While knowledge integration (KI) in terms of influencing factors and knowledge integration outcomes has been examined intensely, the underlying collaborative processes and activities undertaken to integrate different experts' knowledge and create a new product, service or system, are still poorly understood (Tell 2011). In a recent literature review, Tell (2011) shows that, during the past ten years, numerous studies have examined how task characteristics (such as complexity, uncertainty, novelty, and frequency), knowledge characteristics (tacit and explicit, internal versus external, knowledge breadth and depth, degree of common knowledge), and relational characteristics (interaction pattern, social identity, past integration experience, organizational design) affect KI. In addition, how knowledge integration is linked to organizational performance in terms of efficiency, effectiveness and innovation outcomes has also been investigated extensively. On the other hand, the different activities of a specific innovation process that organizational members carry out collaboratively in order to put together their different expertise and develop a new product, system or solution have been examined to a lesser degree. Tell (2011 p.37) concluded that "[t]he lack of dynamic analysis and poor understanding of underlying processes and mechanisms of knowledge integration are two major concerns about the extant literature that need attention in future research."

Grant (1996a) suggested a set of "mechanisms", which entailed rules, routines, directives, sequencing, and group problem-solving and decision-making, which were aimed at facilitating and accomplishing knowledge integration within firms. Grant's mechanisms constitute an important and widely-acknowledged contribution to the understanding of how knowledge integration occurs in organizations. However, these mechanisms are mostly associated with the application and integration of existing knowledge within firms, rather than the integration and creation of new knowledge. As Grant (1996a p.113) stated, "my emphasis is on the firm as an institution for knowledge application." Moreover, current knowledge integration research offers valuable "static" or "steady-state analyses" of influencing factors, performance outcomes, and knowledge integration mechanisms, but there is less attention paid to how knowledge integration is carried out over time by various members in cross-functional development projects or teams (Tell 2011 p.36, 38.). How can we understand the process dynamics of knowledge integration in cross-functional development projects? Engwall and Westling (2004 p. 1559) have studied project dynamics and reported that "projects evolve over time and that a project retains significantly different dynamics in different phases of its life cycle". Söderlund 2004 (p. 189) has argued likewise that "activities vary dramatically over the life of the project, from, e.g. conceptualization, feasibility studies, detailed engineering, to testing and commissioning". As a consequence of this, one might suspect that a certain pre-determined set of knowledge integration mechanisms is not suitable throughout the whole project, but may, indeed, need to be adapted to fit constantly changing situations. This should be especially valid for complex and uncertain development projects that cannot be perfectly specified and planned at the outset of the process since unanticipated challenges and problems repeatedly turn up, which quickly must be collaboratively solved and settled.

In the same vein, Faraj and Xiao (2006 p.1157) suggest a "reorientation of knowledge coordination away from pre-identified interdependences and modes of coordination. This reframing is timely because of the growing recognition that routine coordination (in the sense of recognizable and repetitive patterns) cannot be specified in sufficient detail to be carried out and is, thus, insufficient to coordinate complex knowledge work." How do knowledge integration activities and mechanisms change over the project's lifetime? Majchrzak et al. (2012 p.954) claims that "there is a need in the literature on knowledge integration in teams facing novel situations to move beyond an understanding of formal processes and coordination mechanisms to understand work as a situated activity where synchronization and participation are constantly evolving". How is the combination of various people's knowledge and their contributions in development settings accomplished, and what triggers a change in a particular set, and use of, knowledge integration mechanisms? Söderlund (2010 p.137) suggests a similar view of knowledge integration "as a dynamic process developed in social settings rather than a static form of integrating 'repositories of knowledge' that are just waiting to be used." The present study thus seeks to complement previous research with exploration into the dynamics of knowledge integration and creation and its underlying mechanisms, processes, and activities so as to understand how knowledge integration changes over time in goal-directed development and innovation processes.

One challenge with this undertaking is how one is to accurately and fully capture the dynamics of knowledge integration processes. As long as the development work progresses as planned, and knowledge integration mechanisms and activities seem useful, there is probably no need to make changes in the chosen set of knowledge integration mechanisms. However, previous research has emphasized the uncertainty and complexity inherent in innovation work and has frequently featured the development task as a problem-solving activity or process (Iansiti 1995; Lindkvist et al. 1998; Engwall and Westling 2004; Cross and Sproull 2004). Unforeseen problems constantly turn up which must be solved quickly in order to keep the project within its set limits. Investigating how project members resolve problems and situations that require new paths in terms of altered knowledge integration mechanisms, activities, or collaboration forms may constitute one possible way of looking into the dynamics of knowledge integration.

Exploring the dynamics of knowledge integration involves also a focus on project members in cross-functional teams and how they conduct knowledge integration. Huang and Newell (2003 p.168) state that "cross-functional teams are, in essence, groups which have members with highly differentiated knowledge and a mission that can only be fulfilled through integrating the differentiated knowledge both internally within the group and externally with the various stakeholder groups impacted by the group's work". Majchrzak et al. (2012 p.954) studied knowledge integration by using a similar approach and "focused on the practices used by teams to integrate member knowledge" and framed it as taking a "practice perspective [...] on the work of cross-functional teams to focus on how knowledge integration is actually done". Faraj and Xiao (2006 p.1157) also suggest a "practice view" and argue that it "breaks with perspectives that overemphasize the role of rules and structures at the expense of actors in explaining work activities" and assume further that "practices are driven by a practical logic, that is, a recognition of novel task demands, emergent situations, and the unpredictability of evolving action". Engwall and Westling (2004 p.1572) argue that "there is a substantial amount to learn regarding processes at the level of project participants" and "further research is required that closely aligns itself with how the participants deal with and make sense of the task at hand during project execution." The present study attempts to capture knowledge integration dynamics through focusing on project managers' and project members' problem-solving activities and emerging actions in a cross-functional project from its start to finish.

A final aspect of this research agenda concerns knowledge diversity among project members. While different skills and knowledge domains are needed to solve complex tasks in creative and proper ways, knowledge related differences may easily cause communication- and collaboration problems that hinder innovation (Dougherty 1992; Bechky 2003; Carlile 2002). Thus, as a complex task or innovation mission might require broad imagination, creativity, and the utilization of many different people's knowledge in certain stages, it might demand discipline, convergence and fewer differences in perspectives, interests, and interpretations in other phases in order for the project to progress in a timely fashion. Söderlund (2010 p. 136) states that "[p]roject organizing is therefore, in major part, a matter of identifying who knows what and determining when that knowledge is needed." It is not only important to access task-relevant skills and knowledge, and to integrate this knowledge so as to solve problems and achieve specific goals, but to do so appropriately and orderly over time (Larson 2007).

This study pays attention to how the utilization of differences in expertise, defined as "the specialized skills and knowledge that an individual brings to the team's task" (Faraj and Sproull 2000 p.1555), may need to be managed over time in a particular development process. Larson (2007) notes that coordination of different people's knowledge is difficult to specify in the beginning of a development process; instead, it must be accomplished as the process unfolds. In the same vein, Söderlund (2010 p.136) argues that "project management takes on a role where the widening or narrowing of limits are important, adding or subtracting weights to sort out what trade-offs need to be made, speeding up or slowing down actions, and increasing the emphasis on some activities and decreasing the emphasis on others". He asserts that "the effectiveness of knowledge integration largely depends on how different knowledge processes are tied together, and what time orientations individuals/units have with regard to the integration of their individually held knowledge" (Söderlund 2010 p.133). One might speculate that during "brain-storming", or idea generation phases, more differences in knowledge, skills, interpretations, and opinions might be needed to produce a rich base of creative alternatives; whereas in some later phases, where a defined concept is to be be transformed into a concrete product, less divergence in terms of knowledge, interests, interpretations, and perspectives might be desired. Furthermore, in such late phases of a project, greater effort might be put into synchronizing problem-solving activities and coordinating the different members' contributions in a timely manner, instead of further stimulating the experts to come up with more ideas and create elegant solutions individually.

Against this background, the extended focus of this study will be on developing an understanding of how diversity in expertise is managed in knowledge integration and creation processes. That is to say, how the "space" for applying different skills and knowledge and expressing difference in interpretations, interests, and perspectives is modulated in certain situations or phases by the use of various organizational mechanisms and management tools. It is also of interest to investigate whether project members collectively invent and use a variety of helpful measures to manage and pace knowledge integration and facilitate progression at different stages of the development process.

1.1 Outline of chapter

In part two of this chapter, I start by examining knowledge integration by discussing who conducts knowledge integration; from whose perspective should knowledge integration processes be seen? This involves a discussion of who the research spotlight should be turned to - the person who conducts knowledge integration. I then present two general views on how knowledge can be integrated; through transferring or sharing knowledge, and through the combination of specialized knowledge. I suggest that knowledge integration should be conceived as a matter of *knowledge combination* rather than as one of *knowledge sharing*. This approach sets the stage for determining what is relevant to a more detailed discussion on how knowledge is integrated. Knowledge integration mechanisms and underlying processes and activities are then discussed. Different aspects and understandings of what participants of a knowledge integration process actually do to put their various skills and experience together are also presented. A central line of thought throughout this part is that knowledge integration in innovation and development contexts involves the creation of new knowledge, often manifested in new concepts, products, systems, or solutions.

In part two of this chapter I also propose a location for knowledge integration; the cross-functional development project. The development process that takes place in a cross-functional project corresponds with the way that the knowledge integration process is defined in this thesis. The emergent character of development projects and processes are highlighted, as well as the time-pressure that is commonly put on projects. The development process in projects is further explored and conceptualised as a problem-solving endeavour which includes stages of problem formulation, solution search, and evaluation of results.

In part three of this chapter, I turn to the specific challenge of promoting, constraining, and pacing diversity and creativity over time in a development process. Diversity, in terms of knowledge-related differences is assumed to be vital in innovation and development. Unfortunately though, unique knowledge tends to be suppressed in discussions. There are, however, means to promote and facilitate the mentioning of such knowledge. While differences in knowledge and perspectives are prerequisite to accomplishing creative and novel solutions, it may also cause harmful conflicts and can hinder rapid progress towards task resolution. Thus, the space and time for expressing differences must sometimes be restricted. Different mechanisms and means that will narrow the room for diversity are tentatively suggested. The challenge to balance, that is, to promote and constrain diversity and creativity over a specific development process is also discussed. The aim of the discussion is to understand what mechanisms and measures are in use in different stages of the development process, and if and how the mechanisms change during the course of action. Part four provides a summary overview of the chapter.

2. Conducting knowledge integration

This part of the chapter includes a discussion that takes starting point in the "knowledge integrators", the participants of a knowledge integration process (2.1) and continues with an examination of how knowledge integrators conduct knowledge integration (2.2). Underlying processes and activities are explored. Finally, there is a discussion that considers where these processes and activities may be located in space and time (2.3).

2.1 Who conducts knowledge integration?

Various researchers have argued that knowledge integration and knowledge creation is an individual activity as well as a collaborative process (e.g. Nonaka 1994; Grant 1996a; Brown and Duguid 1998). Few studies have examined what the individuals and teams that conduct knowledge integration actually do (Andersson and Berggren 2011), especially over time from the start to the end of specific development process (Tell 2011). Andersson and Berggren (2011) argue that whereas leaders, project managers, and people in managerial positions to some extent have been recognized in the new product development literature (NPD), working-level engineers have been neglected. "Heavyweight" project managers who organize and manage cross-functional project teams (Wheelwright and Clark 1992), and gatekeepers or boundary-spanning individuals who ensure that appropriate external information and knowledge is acquired (Allen 1977; Tushman and Katz 1980) have been discussed in the literature. However the "doers" have mostly been seen as resources who represent different knowledge domains, and not as important agents in the knowledge integration process (Andersson and Berggren 2011). "When discussed at all, individuals are recognized as sources of knowledge but not studied as potentially important agents in the integration process." (Andersson and Berggren 2011 p.79). New product development is often conceptualised as a process that starts after a goal has been defined and specified, and aims at executing already known tasks and activities, which could constitute one explanation why less attention has been paid to the role that individuals play in the actual work of integrating and creating new knowledge (Andersson and Berggren 2011).

Knowledge integration in technology-based innovation contexts comprises of idea generation and concept formulation in addition to the actual construction and implementation of a new working product or solution. It involves uncertainty and complexity. No one knows exactly beforehand what knowledge will be needed to solve specific problems, what technologies or solutions will work, or what the exact outcome of the process will be. Knowledge integration (KI) in such contexts thus requires exploration and experimentation, problem-solving, and a continuous generation of ideas and solutions, which put strong demands on all participants, not only managers, throughout the entire development process. Andersson and Berggren's (2011) study serves to introduce us to a first understanding of a "knowledge integration-agent" or "innovator". The authors found that successful innovators have skills in several knowledge domains and insight into the whole development process including all its different stages. Typically, a successful innovator also understands the practical use and customer value of the product. In this way, innovators are able to discuss and seek advice from many different specialists and potential end users so as to be able to elaborate on ideas and improve concepts. Informally, knowledge integrators combine the insights and skills from different stakeholders and experts, and then different formal projects are used to test and advance early concepts and solutions. Iansiti (1995) argued similarly that effective knowledge integrators displayed a “T-shape skill pattern”, which means that they had deep knowledge in one domain and broad knowledge of many related areas so that they were able to understand how their knowledge domain related and interacted with other expertise areas as well as contextual factors. Dahlander and O’Mahoney (2011) note how people tend to engage more in coordination work as they advanced laterally in the organization and gained more authority over tasks (but not over people). As the focus in this thesis is on the underlying processes of knowledge integration, attention will be paid to the “doers” or “knowledge integrators” – the participants in innovation processes – to examine how and what they do when integrating knowledge to create new concepts, products, or solutions.

2.2 How is knowledge integrated?

Before I refer to studies that offer insight into underlying processes and activities of knowledge integration, an overall perspective of knowledge integration is presented below. In previous research, at least two approaches have been intensely examined and discussed with respect to how knowledge is integrated; knowledge integration through knowledge transfer and sharing in the sense of “having in common” and knowledge integration through the combination of specialized, but complementary, knowledge. These two perspectives will be discussed in turn since they offer two contrasting perspectives that incorporate differing ideas on what is relevant and worthy of investigation.

2.2.1 Integration through knowledge transfer and knowledge sharing

Scholars who have argued for a ‘knowledge transfer and sharing approach’ have shown in various studies how difficult it often is for different organizational members to communicate, coordinate actions, and collaborate effectively when the organizational members have different backgrounds, education, and experiences (e.g. Carlile 2004, Bechky 2003). Individuals of different functional units of an organization are sometimes described as living in different “thought worlds” (Dougherty 1992), enacting completely different work norms, beliefs, priorities, perspectives, routines, and procedures, which easily results in misunderstandings and conflicts. This takes place even in situations where organizational members are aware of their differences and try hard to collaborate with each other. This is despite the fact that they believe that they have the same picture in mind of what and how something should be done. Unfortunately, only much later, when considerable pieces of a product already have been worked up, do they realize that they misunderstood each other. Precious time and money is then wasted on solutions that might not match customer expectations, or even function at all.

Researchers that adhere to a knowledge transfer and knowledge sharing perspective have stressed the importance of learning from each other to understand and see each other’s perspective and so overcome communication difficulties and collaboration problems. Different means and tools have been investigated and proposed to facilitate the transfer of knowledge between groups and individuals. This is done in order to build a communal base or platform of shared knowledge. In this way, knowledge becomes integrated and interdisciplinary work improves.

One of the ways to spread knowledge across functions and occupational groups so as to integrate knowledge involves the systematic use of two specific roles; translators and knowledge brokers (Brown and Duguid 1998). Translators are people who are skilled in at least two different units' or groups' specialist domains. They are able to mediate between groups and explain one groups' viewpoint in terms of another group's perspective. The translator role is, however, a difficult role to play since the translator must be able to impartially negotiate for both groups and be trusted by both parties. Brown and Duguid (1998) assert that individuals who can enact a translator role are valuable, but hard to find. Translators are often called in from outside the organization; for example, they are hired as consultants. Knowledge brokers, on the other hand, are internal resources and they participate in several groups' or "occupational communities" work, instead of mediating between them. Knowledge brokers facilitate the flow of knowledge among "tightly knit" groups due to the fact that they have "weak ties" (Granovetter 1973) and are loosely linked to several groups and not bounded to any one particular group (Brown and Duguid 1998).

Carlile (2002, 2004) has shown how specialized knowledge can be both a source and a barrier to innovation. This researcher underscores the importance of sharing knowledge through the processes of "transferring, translating and transforming". Inspired by Star (1989), Carlile (2002, 2004) refines and elaborates on the concept "boundary object", which is supposed to enable such processes and collaboration between different stakeholders across knowledge boundaries. A boundary object (e.g. blueprints, prototypes, sketches, notes, drawings, schemas) offers a shared syntax or language, which makes it possible for different people to represent and *transfer* their knowledge. It is also a means to specify and learn about differences and dependencies, and thereby *translate* knowledge. In addition, boundary objects entail an entrance for individuals to negotiate and *transform* prevailing knowledge when problems require novel solutions.

Bechky (2003) emphasizes the importance of transforming knowledge and the creation of "common ground" in order for specialists to communicate and understand each other and be able to take in and assess each other's domain-specific knowledge and solve cross-occupational problems. She argues that knowledge is situated and localised in practice and knowledge sharing across different occupational groups or "communities of practice" (Lave and Wenger 1990; Brown and Duguid 1991) is not, therefore, an easy task. Misunderstandings between different occupational communities are rooted in work contexts, which have differing subtasks, standards, priorities, perspectives, evaluation criteria, and language. According to Bechky (2003) these types of communication difficulties can be overcome by the co-creation of a common ground between the groups. This common ground consists of tangible definitions or concrete manifestations of the problem and the product. This helps transform and enriches the different groups' understandings of the product and situation and so improves cross-functional collaboration and knowledge integration. Huang & Newell (2003 p. 167) also point to the importance of common knowledge for knowledge integration. They claim that knowledge integration can be defined as "constructing, articulating and redefining shared beliefs."

2.2.2 Integration through combination of specialized knowledge

Although the knowledge transfer approach and its appealing logic have offered many valuable insights into the widespread problem of coordinating knowledge across boundaries, this thesis adheres to the stream of research that views and defines knowledge integration as "combining specialized but complementary knowledge" (Tell 2011 p. 27). Okhuysen and Eisenhardt (2002 p. 383) distinguish between knowledge integration and knowledge sharing and suggest that knowledge integration is a process where "several individuals combine their information to create new knowledge", whereas knowledge sharing is a process where "individuals identify and communicate their uniquely held information." Similarly, Söderlund and Tell (2011 p. 171) suggest that knowledge integration concerns the "combination of specialized yet complementary knowledge with the purpose of attaining specific objectives." Grant (1996a p. 113) argues that "[g]iven the efficiency gains of specialization, the fundamental task of organization is to coordinate the efforts of many specialists." It is based on the belief that having different experts in an organization spend time on learning each other's' knowledge domains may not be economically defensible, and even counterproductive, since it means that the economies of specialization and synergetic effects then are not taken advantage of or sufficiently exploited. Demsetz (1991 p. 71) states:

> Although knowledge can be learned more effectively in a specialized fashion, its use to achieve high living standards requires that a specialist somehow uses the knowledge of other specialists. This cannot be done only by learning what others know, for what would undermine gains from specialized learning.

Grant (1996a) argues that the transfer of knowledge between organizational members is difficult, slow, uncertain, and costly. This is particularly true when most of the relevant knowledge is tacit. Whereas explicit knowledge can more easily be communicated between individuals, since it is connected to *knowing about* facts and theories, tacit knowledge is related to *knowing how*, and can only be observed and revealed through its application and acquired through practice. Knowledge integration should therefore not be accomplished through cross-learning or knowledge transfer according to Grant (1996 p.114):

> [T]ransferring knowledge is not an efficient approach to integrating knowledge. If production requires the integration of many people's specialist knowledge, the key to efficiency is to achieve effective integration while minimizing knowledge transfer through cross-learning by organizational members.

Grant states that the firm's primary role is to integrate specialized knowledge while minimizing cross-learning between individuals. Nevertheless, he recognizes the importance of common knowledge and the need for some degree of 'sameness' among organizational members. For example, a common language is needed to be able to perform or follow directives and rules and jointly solve problems. According to Grant (1996a), symbolic communication in terms of raising the knowledge level in, for example, numeracy, literacy, and computer programs facilitates knowledge integration. In addition, reciprocal recognition of individual knowledge domains, that is, knowing who knows what, in order to adjust to each other, is important for effective knowledge integration. Furthermore, a shared understanding in terms of, for example, shared stories, analogies, and metaphors may be important to the communication of tacit knowledge, without losing most of this knowledge when converting it into explicit forms. Nonaka (1994) has asserted that tacit knowledge can be acquired without explicit communication and spoken words. Similarly, Brown and Duguid (1998) discuss "tightly knit groups" and "communities of practice" and underlined the importance of socialization and working closely together to form a common knowledge base and to transfer tacit knowledge. However, this takes time and effort and may result in mere common knowledge than specialised unique knowledge. Lindkvist (2005) criticizes the communities of practice concept and argues that it cannot account for projects and temporary organizations, which "consist of diversely skilled individuals, most of whom have not met before" (p.1189) [...] "who have to engage in swift socialization and carry out a pre-specified task within set limits as to time and costs" (p.1190). Project members' knowledge differences make it also "difficult to establish shared understandings or a common knowledge base" (Lindkvist 2005 p. 1190). He proposes instead the concept "knowledge collectivity" to describe such work contexts and argues that knowledge integration is achieved through "[w]ell-connectedness of knowledge bases" rather than "[k]knowledge base similarity" (Lindkvist 2005 p.1205)

Nevertheless, some commonality of specialized knowledge must exist since completely separate knowledge bases are difficult to integrate. Knowledge integration requires "additivity between different elements of knowledge" (Grant 1996a p. 111). The different knowledge bases must be complementary and common knowledge is needed to some extent since it affects the absorptive capacity of individuals, that is, individuals' ability to add new knowledge to their existing knowledge. However, Grant (1996a p.114) argues that "[t]he general issue is devising mechanisms for integrating individuals' specialized knowledge." The focus is thus not on how to reach high levels of common knowledge. Instead, knowledge integration should be achieved by the use of different knowledge integration *mechanisms* that allow the combination and integration of individual knowledge bases with minimal effort spent on cross-learning among participants. This entails that organizations can choose from a set of generic means for using and integrating individuals' specialized knowledge. This will be further explained in the next section.

2.2.2.1 Knowledge integration mechanisms

Grant (1996a) suggests four different knowledge integration mechanisms. These are "rules and directives", "sequencing", "organizational routines", and "group problem-solving and decision-making". Basically, the first three are associated with little or no interpersonal communication and learning, whereas the fourth mechanism relies on personal, unregulated interaction, and face-to-face communication. Knowledge integration through interpersonal communication will be discussed in more detail below in section 2.2.2.3 since it is assumed to be a major form of interaction and a knowledge integration device in complex development settings. The first three constitute "cheap" integration mechanisms since they economize on interpersonal communication and interaction, while the fourth mechanism is far more expensive, involving the integration of hard-to-communicate explicit, as well as tacit, knowledge. Whereas contingencies, such as minimal interdependencies and limited complexity, allow for the use of mechanisms such as rules, roles, and routines, severe "team interdependencies" (Van de Ven et al. 1976) call for expensive and intensive communication mechanisms, such as "group problem-solving and decision making" (Grant 1996a p. 114). Choosing appropriate organizational mechanisms from such a "menu" should thus solve the knowledge integration issue.

Schmickl and Kieser (2008) studied three different, but interrelated, knowledge integration mechanisms and their impact on reducing the need for transferring knowledge among specialists. These mechanisms are transactive memory (Wegner et al. 1991), modularization, and prototyping. Their research showed that transactive memory, which can be compared to Lindkvist's (2004 p.15) network memory, which involves knowing "who knows what", constituted a shortcut in the search for specialists, and required knowledge outside and inside the organization. This thus reduced interpersonal communication and verbal exchange among individuals on, for example, the task complexity, problems, needs, and requirements. Modularization is meant to break down a large and complex task to smaller and simpler subtasks or components. It allows specialists to work independently on their own subtasks. Modularization thus reduces the need for knowledge exchange and coordination among experts. However, the researchers observed that some communication was still needed, but it was mostly regarded as being a result of the components' interaction and interface difficulties. The communication observed in the study was typically problem-driven and featured a "question and answer discussion pattern" where interface details and problems were collectively analysed to ensure smooth interaction of components. The elements of each component were, to a great extent, not relevant to the discussion, and, therefore, were left out. The different specialists concentrated on the interface problems and issues. As an aid in this process, prototyping was commonly used. With the prototypes, the experts could identify specific challenges and problems in component interaction. As more pieces were developed and put together, the prototypes included gradually more and more information and became more and more sophisticated. Even though the mechanisms reduced cross-learning, as the researchers explain, there was still a greater need for iterative problem-solving, interpersonal communication and hence knowledge transfer, , between the specialists in radical innovation than among those conducting incremental innovation. This is due to the fact that complex interface design is difficult to set and fix in early phases when all the relevant knowledge does not yet exist.

Okhuysen and Eisenhardt (2002) carried out laboratory research on knowledge integration in groups by using an experimental design. They found that simple formal interventions or mechanisms such as "managing time", "questioning others", and "information sharing" can affect and often improve knowledge integration when participants have different knowledge bases. To be exposed to time pressure and to be aware of the time aspect and focus on time-pacing seemed connected with effective problem-solving and task completion. In addition, to ask others about their knowledge will make individuals' unique knowledge more easily accessible to the group and will thereby enhance the effectiveness of problem-solving and knowledge integration. These formal interventions also caused members to reflect on how they carried out their task, which resulted in changes in the way of working (changing speaker, way of talking, content focus) during the course of development and thus changed how knowledge was integrated. Okhuysen and Eisenhardt (2002) argue that the participants created a "second agenda", and that it was the primary effect of formal intervention on knowledge integration performance. The formal invention, "information sharing", was expected to encourage members to communicate their uniquely held information. Interestingly enough, however, it did not improve knowledge integration. It merely resulted in a self-oriented focus; the participants concentrated on themselves and their own chances to talk and to communicate what they knew during discussion. Söderlund (2010) and Lindkvist et al. (1998) have also researched the importance of time management, milestones, and deadlines for knowledge integration in projects. Creating a sense of urgency among project workers may stimulate a certain type of "global thinking" and creative reflection – a process where the individual relates his or her actions to a wider context – and communication across functions.

Faraj and Sproull (2000 p. 1556-1557) discuss process dynamics in their research into expertise coordination in software development teams. They argue that the mere existence of the "right" expertise is not enough to ensure good performance – the expertise must also be carefully coordinated through mechanisms such as recognition of who knows what, that is to say, "Knowing Expertise Location", understanding when and where that knowledge is needed, "Recognizing the Need for Expertise", and organizing for smooth informal interaction to access relevant expertise, "Bringing Expertise to Bear". The authors also mention other administrative coordination mechanisms and tools that can be more easily pre-specified, such as formal software development methodologies, milestones, and review meetings. Similar to “Knowing Expertise Location”, Tiwana and McLean (2005 p.33) discuss “Relational Capital” as an important mechanism in expertise integration processes and claim that the “accessibility of other individuals’ expertise within a team is an important predictor of its application to the project, especially when a detailed breakdown of each member’s contributions cannot be fully anticipated in advance.” According to Tiwana and McLean (2005) “Absorptive Capacity” was also critical for expertise integration, creativity and knowledge generation. As Cohen and Levinthal (1990 p. 133) reason, “[a]ssuming a sufficient level of knowledge overlap to ensure effective communication, interactions across individuals who each possesses diverse and different knowledge structures will augment the organization’s capacity for making novel linkages and associations – innovating – beyond what any one individual can achieve.”

Faraj and Xiao (2006) and Majchrzak et al. (2007 p.156) argue that coordination in fast-response organizations and emergent response groups is achieved in a different manner. In such organizations and situations, roles, tasks, responsibilities, membership, and circumstances change rapidly, which makes “metastructures” on who-knows-what quickly outdated (Majchrzak et al. 2007). Here “expertise coordination practices” such as “plug-and-play teaming” and “dialogic coordination practices” like “joint sensemaking” (Faraj and Xiao 2006; Majchrzak et al. 2007) and “running narratives of the actions taken and not taken” (Majchrzak et al. 2007) evolve instead as coordination mechanisms, or rather, coordination processes and activities.

The coordination mechanisms that have been presented above constitute just an introductory list of knowledge integration mechanisms that can serve as a starting point in the reading of the empirical case presented in this thesis. Many other studies could have been included here too. The studies that were selected and commented on above all have one or more of the following features. They defined ‘knowledge integration’ in a way that is similar to what has been done in this research, *or* they considered process dynamics when discussing coordination, *or* they focused specifically on knowledge and expertise integration (and not merely on coordination or integration of information, tasks or actions in general), *or* they discussed integration and coordination in groups or project teams rather than in or between entire organizations. The discussion on knowledge integration mechanisms thus provides us with one basic point of departure for the dissertation, since it helps us to identify mechanisms that are used to integrate different specialists’ contributions.

While a strong foundation of common knowledge may facilitate collaborative effort, my research interest on investigating the “opposite” question of how a diverse set of knowledge bases may be efficiently connected, without heavy investment in developing a strong base of communal knowledge and shared understandings. Grant’s “mechanism menu” is valuable in a contingency analysis that aims to provide understanding of how each of the mechanisms may be appropriate in a specific context. However, as Grant states, his framework explains knowledge integration in the context of knowledge application and is not primarily developed to account for the simultaneous integration and creation of new knowledge that takes place in development projects (Grant 1996a). Furthermore, the “menu” suggests that the mechanisms are, in a way, structural devices rather than “process drivers” in the knowledge integration process that supports problem-solving and makes sure the process proceeds in a timely manner. How knowledge integration mechanisms are developed, used, and how they change during an ongoing specific development process characterized by collaborative problem-solving and creativity has not yet been widely researched. Okhuysen and Bechky (2009) offer a literature review on coordination and coordination mechanisms, starting out in classic organization theory. Bringing their historical overview to the current period, they conclude that coordination in contemporary firms should be better understood and researched as an “ongoing accomplishment in organizations” (Okhuysen and Bechky 2009 p.493) since “coordination is under persistent attack by the regular dynamics of organizations” (Okhuysen and Bechky 2009 p.494). Future studies should focus on integration processes and necessary conditions for integration to occur rather than merely looking at how the mechanisms that should help achieve coordination operate (Okhuysen and Bechky 2009). This point is discussed more thoroughly in the next section. Even if there is a lack of such studies mentioned above, some valuable contributions to the understanding of knowledge integration processes and activities have been made.

2.2.2.2 Knowledge integration processes - activities

According to Tell (2011), there is a shortage of studies that focus on activities, behaviour, and dynamic underlying processes of knowledge integration in innovation contexts, especially longitudinal studies. Research on how participants who are involved in knowledge integration use and combine their different skills and experience in the course of a specific development process has been neglected. However, there are some studies that give inspiration as to what this may entail. As a point of departure, I have chosen to use recent empirical studies which have investigated what individuals in development contexts or unpredictable high-speed environments actually do to integrate their diverse perspectives, create new knowledge, and solve various problems without investing a large amount of time in cross-learning activities.

Majchrzak et al. (2012) identify five different practices that cross-functional teams used so as to integrate the team members’ various knowledge domains in novel problem-solving processes. Faraj and Xiao (2006) discuss two practices that each contained different activities or subprocesses which enabledcoordination over time in fast-response organizations, as mentioned above. Okhuysen and Bechky (2009) also emphasize the dynamic and ongoing character of coordination and pay specific attention to three conditions that enable coordinated activity. Söderlund (2010) researched large-scale transformation projects and discusses timing and synchronization of knowledge integration processes. Enberg et al. (2006) examine knowledge integration in product development projects. Lindkvist et al. (2011) researched knowledge integration and knowledge creation within project contexts, and Hargadon and Bechky (2006) investigated problem-solving and collective creativity. These three studies state in general that integrating knowledge is both an individual process that takes place within an innovation or development context, as well as a collaborative process where experts integrate and create knowledge in interaction. Kellogg et al. (2006) and Matusov (1996) report on evolving coordination processes in terms of an “emerging collage” and progression that falls out “mosaicly” and underscored the finding that activity *can* be coordinated and knowledge integrated in spite of differences in perspectives among participants.

These studies offer a number of different descriptions, explanations, and insights into the dynamic underlying processes of knowledge integration. They were chosen to guide the analysis and detection of what KI mechanisms and processes were in use in the case project studied in this thesis. These studies also serve as a searchlight for detecting other practices, activities, sub-processes, and means or enablers that assist in problem-solving and help accomplish knowledge integration in cross-functional projects. I will also try to understand how these kind of processes and activities may change over time in a specific project development process, which current research does not reveal (with the exception of e.g.

Majchrzak 2012 and Faraj and Xiao 2006). But before we pay more attention to each of these studies, I will present a discussion of knowledge integration as a process at a more general level.

Berggren et al. (2011 p.8) suggest a model of knowledge integration as a process that contains three elements: *Inputs → KI → Outputs*. I will adhere to this model in defining this concept. *Inputs* of a knowledge integration process consist of goals or intended outcomes of a particular development or innovation undertaking. It also includes the different knowledge bases that specialists bring to the process.

KI involves the underlying processes that combine these knowledge bases and, importantly, create the new knowledge that the task at hand may require to be accomplished. "KI is not only a process of combining and fusing different knowledge bases but also a process of creating new knowledge needed for this integration to succeed", as Berggren et al.(2011 p. 7) put claims. What the outcome of a complex development process will be is unknown at the beginning of the process. Participants may not have all the relevant knowledge at the outset and do not even know what will be relevant knowledge since the task is, at least, to some extent, new and relatively unique. The process involves concept development. This includes the formulation and definition of what is to be integrated in the new product and how it should operate, as well as development and implementation (including product-, system-, or process design solutions), manufacturing, and testing. The organization of the knowledge integration process may vary and change during a specific course of development, as well might the contextual and organizational factors that influence knowledge integration.

Output concerns the physical or informational result of the process. It consists of products, processes, systems, services, or solutions. The end result is then evaluated in relation to the goal or intended outcome. Since KI has been defined as a process at an overall level, we can now turn to the studies that were mentioned earlier to gain insight into different underlying knowledge integration processes, activities, and communication.

Majchrzak et al. (2012 p. 963) suggest that knowledge differences in cross-functional teams which may hinder knowledge integration and innovation can be transcended through five different practices that, together, facilitate integration of diverse knowledge and creation of solutions to novel problems. The practices are:

i. Voicing fragments ("assembling a common landscape of individual statements and parts of solutions")
ii. Co-creating a scaffold (developing a preliminary "fluid" representation which offered a common elaboration experience)
iii. Dialoguing around the scaffold (surfacing hidden tensions which stimulated creative solution generation)
iv. Moving the scaffold aside (further co-creation to integrate external stakeholders' requirements), and
v. Sustaining engagement ("through repeatedly summarizing, the sharing of the unexpected, and the use of collective enthusiasm").

These practices helped overcome knowledge integration hindrances without engaging in confronting, clarifying, and resolving knowledge differences. As Majchrzak et al. (2012 p. 963) argue, "[t]he practices more specifically depict how knowledge integration challenges are overcome over time as a team goes from individuals representing specialist knowledge areas to the creation of a collectively integrative solution. The practices describe how sensemaking evolves, how the actions of previously unknown others become anticipated, how members create a psychologically safe environment to engage in iterative and rapid reflection, how creative breakthroughs occur without creative tensions between individuals, and how knowledge transformation occurs between different languages and perspectives without deep-knowledge dialogue."

By studying high velocity environments (trauma centers), Faraj and Xiao (2006 p. 1160) also identified some "coordination practices", as was briefly mentioned in the previous section. They call these practices "expertise coordination practices" and "dialogic coordination practices". The first practice "refers to processes that manage knowledge and skill interdependencies" and included "reliance on protocol" (which streamlines work and reduces uncertainty), "plug-and-play teaming" (which allows for flexibility to meet changing requirements), "communities of practice" (with responsibility for "scheduling, training and control") and "knowledge externalization" (which involved sharing patient information and treatments). The second practice, the "dialogic coordination practice", stems from situations where a deviation or problem has turned up and which requires quick alternative solutions and actions. Faraj and Xiao (2006 p. 1164-1165) explain that this practice includes "epistemic contestation" (triggered by "different beliefs among different specialists as to which treatment step is required"), "joint sensemaking" (triggered by a situation where the patient does not respond well to treatment), "cross-boundary intervention" (triggered by a team member violating a patient's safety), and "protocol breaking" (triggered by the insight that following the protocol might affect treatment negatively). The coordination effect of these practices is suggested to be associated with the overarching goal of patient safety and survival.

Okhuysen and Bechky (2009) reviewed the literature on coordination in organizations and suggest three integrating conditions for coordination to take place: accountability (making responsibilities of interdependent members clear), predictability ("having a sense for what subtasks make up larger tasks and in what sequence tasks will be performed" p. 486), and common understanding (sharing "knowledge of the work that is to be done, how it is to take place, and the goals and objectives of the work" p. 488). Okhuysen and Bechky (2009 p. 483) explain that "[c]entral to our framework is the idea that each condition can be accomplished through a variety of mechanisms. Thus, when individuals respond to the demands of integrating specialized work, they do so by enacting different mechanisms that create the integrating conditions for coordination, drawing from a wide variety of options to achieve them." For instance, accountability can be accomplished through rules, roles, and routines, and predictability might be achieved through plans and time mechanisms. Finally, common understanding may be created through emergent interaction and objects. These authors argue that the three integrating conditions can be seen as "intermediate constructs between coordination mechanisms and coordinated activity" and that their framework captures the wide range of mechanisms presented in previous research and focuses on "what those mechanisms accomplish…rather than relying on the presence or absence of a particular mechanism to explain how coordination occurs" (Okhuysen and Bechky 2009 p. 492).

Both Okhuysen and Bechky (2009) and Faraj and Xiao (2006) emphasize the emergent character of coordination work and assert that coordination is something that takes place and unfolds as individuals solve problems and resolve different situations and perform the work – instead of being a structural form or mechanism that can be pre-determined and specified in advance. Okhuysen and Bechky (2009 p. 493) state that "[n]ormal, everyday dynamics in organization intrude on coordination by eroding the integrating conditions. In these cases, the integrating conditions must be reestablished for successful coordination to occur." In my interpretation, this suggests that coordination mechanisms might need to change over time as a response to different problems and changes in situations that break with the prevailing order of coordination and integration.

Söderlund (2010) researched large-scale transformation projects and discussed a dynamic process view of project management and knowledge integration. He proposed a "knowledge entrainment" model in which timing, pacing, and synchronization of different knowledge processes were essential elements. Different mechanisms, such as visionary meetings, exhibitions, pilot implementations, co-location, and a sense of urgency were used by the project management to orchestrate, pace, and synchronize the different members' contributions and problem-solving activities. "By pointing to various measures taken by the project management we argue that many of these activities are actually ways of stimulating and orchestrating knowledge integration and implementing a sense of urgency generally produced by a challenging deadline." (Söderlund 2010 p. 140). This gives me inspiration and awakens further questions as to whether time-based control mechanisms are used to speed up knowledge integration at the expense of creativity in solutions and specialized problem-solving efforts, and how the project management comes to make such 'trade-off' decisions throughout a project process.

Enberg et al. (2006) uses a model of dynamic knowledge integration to show how team members in product development projects iteratively worked alone and worked together. When working alone, project members integrated knowledge tacitly. When they were working together, knowledge was integrated more explicitly. Enberg et al. (2006) found that project members needed formal routine-based project meetings to receive information, to inform each other, and to ensure that one's own contribution fitted in with "the whole". Sometimes, unexpected events and problems turned up that also required interpersonal communication to be resolved. This implied that project members put some effort into articulating and exchanging knowledge explicitly. Yet when project members worked individually and acted alone, knowledge was integrated with the help of a specific artefact of the product. The project members contributed to the task achievement (more) tacitly, and, in this way, they saved the project cost and effort, since they then cut down on face-to-face meetings. The researchers show that, in this way, complex product development sometimes requires dense and frequent interpersonal communication, which is line with previous research (e.g. Wheelwright and Clark 1992), but that individual action can replace face-to-face communication and be important to knowledge integration since it economizes on expensive and time-consuming interaction.

Lindkvist et al. (2011) emphasize the collaborative nature of knowledge integration and knowledge creation processes. New knowledge arises when diverse people meet and interact and blend their domain expertise, ideas, and perspectives. New knowledge will generally be generated in product- or systems development projects. In a development context, which involves problem-solving and the recombination of diverse knowledge in order to achieve a specific goal, knowledge integration and knowledge creation can be understood as "two sides of the same coin" (Lindkvist et al. 2011 p. 61). Three underlying dynamic processes of knowledge integration and creation are suggested: *preselection*, *variation*, and *selective retention*. Project team members need and use each other to formulate a project vision and project goal, which presumably will change during the development process. In a way, the participants of the process collaboratively create a point of departure, a "searchlight" to guide the development process. This process is called *preselection*. Project members are also assumed to generate new ideas and conjectures by acting on each other's knowledge in an improvisational mode as they learn from experiments and trials.

The outcome of the development process is unknown at the outset of the process and indefinite during the ongoing process. It is also difficult to trace backwards and explain retrospectively; the result emerges "unfathomably" in the interaction and communication among team members. This process is called *variation*. The third activity in the knowledge integration and creation process, *selective retention*, has to do with the assessment of ideas and achievements. Individuals generally have difficulties in evaluating their own ideas critically. They engage, therefore, in collective critical inquiries, for example, different tests, prototypes and tollgates, in order to judge, select proper alternatives, and make decisions. Altogether, one may say that interaction is possible also among diverse individuals. It drives knowledge integration and creation – especially in situations where members do not transfer or share the same knowledge. The members do not even know how their contributions or knowledge will be interpreted by others, where the knowledge will be taken, how it will develop, or what form it will take on in the end.

Hargadon and Bechky (2006) studied collective creativity in organizations by investigating collaborative problem-solving among individuals in six professional service firms. They found that the moments that triggered creativity and novelty in problem-solving involved four types of interaction or collaborative activities: (i) *help giving*, (ii) *help seeking*, (iii) *reframing*, and (iv) *reinforcing*. The core action in collective creativity is *reframing* because it involves drawing on different people's experiences and ideas to critically reflect on a certain problematic situation and reformulate the initial question and problem definition. This is done in order to see it from a new angle and then redirect the search for a solution. According to Hargadon and Beckhy (2006) the culture of the different organizations that were studied encouraged "help seeking" and "help giving" activities or behaviours through reward and incentive systems, supportive principles in the organizations' handbooks, and by the fact that managers allowed their team members to informally and freely participate in other projects and teams. The help seeking and help giving behaviour was also reinforced by positive experiences from such problem-solving meetings. Here, members were able to seek help and give help spontaneously without intense preceding socialization or close relationships. Organizational members could also assist in reframing a problem which they had not been working on previously. Just as in Lindkvist et al. (2011), Hargadon and Beckhy's (2006) study shows that diverse skills and experiences, rather than common knowledge, are crucial in collaborative problem-solving and collective creativity. In this case as in Lindkvist et al. (2011), participants could not always know beforehand how their knowledge could contribute to the solution of a particular problem.

The idea of achieving knowledge integration based on a limited base of shared knowledge is also explored in Kellogg et al. (2006) who conducted an empirical study of a firm that created web-based interactive marketing solutions. In their view, traditional boundary-spanning activities, such as transfer, translation, and transformation of knowledge using artefacts like boundary objects, a common lexicon, common language, and different kinds of knowledge integration roles (Carlile 2002, 2004; Brown and Duguid 1998) tend to be less effective in dynamic, heterogeneous, and decentralized settings. Such activities, artefacts, and roles involve deep investment in terms of agreements and consensus, shared protocols, and the elaboration of objects that run the risk of soon being ineffective and even useless in dynamic and loosely coupled settings where boundaries, values and knowledge are continuously changing. Instead, Kellogg et al. (2006) take a point of departure in Galison's (1997) concept "trading zone" to envision how exchanges and the coordination of actions may be accomplished despite differences in norms, meanings, and values across functional communities, and instable work constellations and relations. The trading zone permits coordination through the agreement on general procedures of exchange. Trading participants from different groups do not need to share understandings; they can still carry their local interpretations of the problems and the task, and can even have different views on the exchange itself. Fiol (1994) also shows how individuals can hold on to different perspectives and interpretations and still agree on how to frame their communication. This researcher argues that simultaneous agreement and disagreement is important in learning and innovation.

Kellogg et al. (2006) identify three types of activities that are mediated by digital technologies; *display, representation,* and *assembly*, and explain how these can be understood as coordinating practices. By using digital technologies such as the intranet, email, a digital calendar, and standard project genres including how to utilize PowerPoint presentations and Word and Excel documents, organizational members displayed work across boundaries and made their past and present material visible and accessible to all organizational members. The work was represented and expressed in a form that was legible to other groups. Different groups could access the material, and reuse, recombine, and assemble it in new ways in different projects across time and space; "they assembled their separate contributions across boundaries into an emerging collage of diverse elements" (Kellog et al. 2006 p. 39) Instead of transferring knowledge, participants make their work visible in a digital space that other groups can access. Rather than merely translating knowledge, actors make their ideas legible to others. Instead of transforming knowledge into shared meaning, they put their different contributions together "into a provisional and emerging collage of loosely coupled contributions" (Kellogg et al. 2006 p. 38). Kellogg et al. (2006 p. 39) view cross-boundary coordination as "performative, as emergent in recurrent actions, and thus as a provisional and ongoing accomplishment."

Finally, Matusov (1996) also argues that there has been an overemphasis on intersubjectivity in terms of the overlap of individuals' understandings and agreement, consensus seeking, and resolution of conflict in joint activity. For Matusov (1996), intersubjectivity and the traditional metaphor of sharing as "having in common" implies an activity or process of unifying or standardizing the participants' understanding and contributions. She argues that intersubjectivity in that sense can actually destroy coordination and joint development. Intersubjectivity should be seen as an agreement of "multiparty coordinated action" instead of just cognitive agreement of the interpretation of the situation. Disagreement is equally important for joint activity. In a similar way, a prosecutor and a defender in court disagree; they do not normally seek consensus but need and depend on each other to develop the argumentation and their contributions, which the judge and the jury coordinate during the process. Similar to Kellogg's et al. (2006) "emerging collage", Matusov (1996) refers to the coordination process as a progression that occurs in "mosaic fashion". She explained that contributions to the whole are not necessarily rule-based, sequenced, or planned exactly in advance. Instead, when participants (in her study) "drifted and jumped" between activities and altered focus in a flexible and dynamic way, they tended to find and explore new possibilities, new goals, and new ways of reaching them. Members did not see the whole picture or took the entire ongoing activity into consideration. Nevertheless, there was still a progression in the activity. This way of coordinating actions is rather different from coordinating and integrating actions through intersubjectivity and knowledge sharing. Matusov (1996 p. 33-34) argues that the conventional way of looking upon intersubjectivity emphasizes stability and repetition;

“intersubjectivity as sharing stresses reproductive aspects of learning[…]at the expense of their productive creative aspects […]. It is very difficult to use this notion to describe how something new develops in a joint activity.”

To summarize this section, I will refer to some ideas and contributions from the studies mentioned above that are specifically relevant to the reading of the empirical material. First, one can understand and conceptualize knowledge integration as a dynamic process and “ongoing accomplishment” (Okhuysen and Bechky 2009) consisting of different practices, sub-processes, and activities (e.g. voicing fragments, plug-and-play teaming, epistemic contestation, pacing and synchronization, acting vs. interacting, pre-selection, variation, selective retention, reframing, help-giving, help-seeking, display, representation, and assembly, mosaic progression) that different members carry out while solving problems, coordinating their various contributions, and performing their work. In such a view, different knowledge integration mechanisms constitute structural devices (e.g. rules, plans, deadlines, sequencing, routines, roles, objects, and an emerging collage) that are used to organize and set up conditions for this “ongoing coordination accomplishment”. As team members perform their work and integrate knowledge, “everyday dynamics” (Okhuysen and Bechky 2009), problems and different challenges influence and interrupt the working order and the integrating conditions (Okhuysen and Bechky 2009) which implies that the members might elaborate new practices and activities to reestablish the order to make the knowledge integration process work. This may also mean that coordination or integration mechanisms must change over time as a response to problematic situations and evolving challenges. This state of affairs is illustrated in the figure below.

The large blue arrow represents the overall knowledge integration process. It contains several practices, sub-processes, and activities which help achieve knowledge integration through the creation of integrating conditions. The knowledge integration mechanisms are structural elements that assist in shaping the conditions for the practices and sub-processes so that they can take place. Due to emerging situations and problems, the mechanisms might need to change over time to maintain suitable integrating conditions, knowledge integration activities and processes.

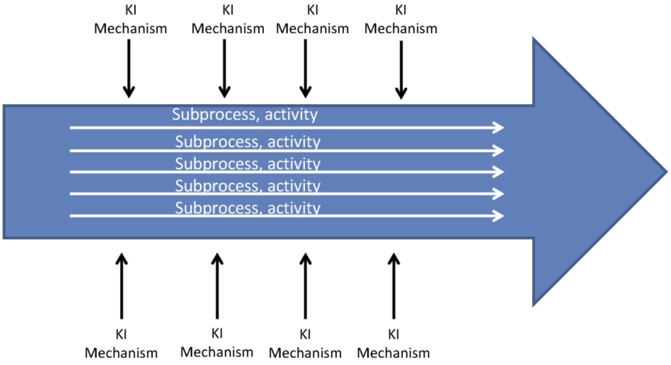


Figure 1: The Knowledge Integration Process

Hargadon and Beckhy (2006) found that collaborative activities were possible and important without strong ties, socialization, or common knowledge. The “reframing” of a situation and problem collectively seemed especially powerful in problematic situations where current search or solution leads nowhere. Kellogg et al. (2006) suggest the concept ‘trading zone’ with practices used to exchange knowledge and integrate contributions to collectively create solutions to problems. The collage metaphor reminds one of Majchrzak’s et al. (2012) “early phase” activity “voicing fragments” and is appealing in understanding how new solutions turn up and grow without sharing specialized knowledge. Similarly, Matusov (1996) discusses “multiparty coordinated action” and dynamic coordination which occurs progressively in a mosaic fashion, without intersubjectivity being present among participants.

Majchrzak et al. (2012 p. 963) elaborate on a sequence of practices that show how knowledge integration can be accomplished over time as a team goes from “individuals representing specialist knowledge areas to the creation of a collectively integrative solution.” In Faraj and Xiao’s (2006) study, the most interesting thing related to accomplishing knowledge integration over time was the finding that some practices (expertise coordination) were regularly used to integrate knowledge, whereas others (dialogic coordination) were enacted as the participants (and the patient) faced unanticipated problems and severe challenges. In reading and analyzing the empirical case of this research, I bring with me the insight that specifying in advance which practices and structural mechanisms will be most appropriate in coordinating work and integrating knowledge seems difficult due to “everyday dynamics”, challenges, and problems that intrude upon work processes and force changes in mechanisms and sub-processes to re-establish integrating conditions (Okhuysen and Bechky 2009, Faraj Xiao 2006). This raises more questions about how these elements (knowledge integration processes, activities, mechanisms, everyday dynamics, unanticipated problems, and challenges) work, change, and relate to each other over time in a specific development project.

Söderlund’s (2010) study inspired me to further examine how timing, pacing, and the synchronization of different sub-processes or activities can be managed over time, as a specific project unfolds. Enberg et al. (2006) reminds one of how important it is to “economically” integrate the tacit aspect of specialized knowledge. This standpoint, when connected with Söderlund’s (2010) ideas, raises questions about how to orchestrate and integrate such knowledge. Moreover, if one looks at a project as a process that consists of three collaborative processes, *preselection*, *variation*, and *selective retention* (Lindkvist et al. 2011), then one might wonder which mechanisms are useful for each process. Furthermore, how do project members and management know when enough has been done in each of the processes or phases and when it is time to move on? A similar question is raised by Söderlund (2010); How can they manage different knowledge processes in a timely fashion? If these processes can take place at the same time or if project members go back and forth between the processes, then one might ask: How can these sub-processes be synchronized?

In the next section, I present a discussion of knowledge integration processes in terms of interpersonal communication. Different types of conversation that are particularly important for problem-solving and the creation of new knowledge are introduced. At a general level, face-to-face interaction or “group problem solving and decision making” (Grant 1996a) can be conceptualised as a knowledge integration mechanism in structural terms or as an organizational design principle. However, the communication exchange, the dialogue that takes place within such interaction is probably better understood as a sub-process that is intended to assist in the achievement of knowledge integration. To a large extent, my empirical material consists of meetings and conversations, and so we might reasonably expect that something of interest for the understanding of knowledge integration will be found in said conversations. We will proceed to go a little deeper into such conversations to form an understanding of what might occur in such exchanges.

2.2.2.3 Knowledge integration processes - interpersonal communication

Many researchers have asserted that unregulated, informal (as well as formal) face-to-face interaction and intense verbal communication play a crucial role in new product development and knowledge integration (Pinto & Pinto 1990; Daft & Lengel 1986; Wheelwright and Clark 1992) and knowledge creation (Nonaka 1994; Tsoukas 2009). Instant and frequent interpersonal communication and interaction is necessary to explore new terrains effectively and solve unforeseen problems quickly. For example, as suggested by Li et al. (2007), information should flow easily and directly between specialists who are involved in complex problem-solving activities. In their literature review on new product development, Brown and Eisenhardt (1995 p. 358) argue that "high internal communication increases the amount and variety of internal information flow and, so, improves development-process performance." Furthermore, Wheelwright and Clark (1992 p. 180) state that "communication that is rich, bilateral, and intense is an important, even essential, element of integrated problem solving". Indeed, ever since the 1980's, project scholars, and new product development and knowledge management researchers have stressed the importance of co-located interaction and communication in multifunctional development and innovation settings (Pinto & Pinto 1990; Daft & Lengel 1986; Sapsed & Salter 2004; Wheelwright & Clark 1992; Hedlund 1994). The more uncertain and complex the task, or the more crisis-like the situation, the more group problem-solving and face-to-face interaction (as a knowledge integration mechanism) is needed (Grant 1996a). Thus, this should be the major mode of interaction in complex development settings.

What is goes on in face-to-face interaction and communication? Engaging in creative dialogue involves, according to Nonaka (1994), participants who share ideas freely and frankly, give constructive criticism based on logical arguments, permit different perspectives, and always view the current conversation as temporary and open for revision. Participants articulate their ideas, theories, and hypotheses and test and verify them in communication with others. Detailed or "redundant" information is important since it facilitates collaborative problem-solving by making it easier for people to understand each other's problems. Tsoukas (2009) also discusses the significance of relational engagement and personal attitude with respect to productive dialogues, commenting on how individuals take responsibility for the joint task and the relationships they have with other participants.

Teasley (1997) and Berkowitz (1983) found that a certain type of conversation, conversations which revealed "transactive reasoning", enhanced and speeded up joint problem-solving. The core of this type of conversation is concerned with the production of a specific sort of utterance, called *transacts*. A transact is a statement that elaborates, defines, declares, refines, criticises, raises a question, or summarizes something. In a problem-solving conversation, participants should use their conversational turn to produce transacts and so embrace what the other participants just have said and mindfully add something "new" to the conversational content and problem-solving process. Conversations that display this type of collaborative reasoning show improved problem-solving, and solutions were of a better quality and accomplished faster than conversations that displayed non-transactive reasoning processes. Okhuysen and Eisenhardt (2002) emphasize their claim that "other-directed" communication also improves knowledge integration, whereas "self-directed" talk does not quicken or enhance problem-solving and knowledge integration.

New knowledge, in terms of new concepts, is created in communication between individuals, according to Nonaka (1994). Tsoukas' (2009) also argues that new knowledge is created through conceptual change in face-to-face dialogues. According to Tsoukas (2009), conceptual change occurs when participants distance themselves from their in-built way of understanding and acting, and reformulate a situation and expand, reframe, or combine the current concept with other concepts. Nonaka (1994) stresses the importance of analogies and metaphors in interpersonal communication. Using metaphors implicitly implies that a concept can be understood and expressed in terms of another concept. Nonaka (1994) explains that it is a way to explore, reveal, and express things, often in images, that are difficult to describe literally. Metaphors and analogies are used to share and merge images and different perspectives. By using metaphors, tacit knowledge can be converted into explicit knowledge, thus this type of expression can be seen as important devices in the "conceptualization" process, which is a cornerstone in Nonaka's (1994) "externalization" and knowledge creation process. Analogies help individuals see similarities between disparate things, and, together, metaphors and analogies constitute a way to enhance understanding of the "future" and the "unknown". Most interesting to note here is the inherent creative power that resides in metaphors and analogies (Nonaka 1994; Tsoukas 2009). It is also important that people engage in other-oriented dialogue (Okhuysen and Eisenhardt 2002), voicing fragments (Majchrzak et al. 2012) and collaborative "transactive" reasoning (Teasley 1997) to build on each other's knowledge and co-create an emerging collage (Kellogg et al. 2006) or preliminary scaffolds (Majchrzak et al. 2012), which function as integrative solutions to novel problems.

The last two sections, "knowledge integration processes – activities" and "knowledge integration processes – interpersonal communication" provided the reader a first hint of the underlying processes and activities of KI may entail. These sections will now be summarized.

2.2.2.4 Summing up KI processes – activities and communication

Majchrzak et al. (2012) identify five different practices which are employed to overcome knowledge differences in cross-functional teams and to integrate diverse knowledge. Faraj and Xiao (2006) elaborated on two coordination practices that each entailed different sub-processes or activities which integrated and, at the same time, challenged current expertise and perspectives on the subject. Okhuysen and Bechky (2009) argue that knowledge integration mechanisms accomplish three integrating conditions which are necessary for coordinated activity to occur. Both Okhuysen and Bechky (2009) and Faraj and Xiao (2006) underscore the dynamic nature of knowledge integration and coordination, and suggest that it should be seen as an evolving ongoing accomplishment. Söderlund (2010) also emphasizes the dynamic character of knowledge integration, but discusses it in terms of the demands that are placed on the project management team to pace and synchronize different knowledge processes. Enberg et al. (2006) shows how a KI process can be understood as a dynamic process, and discusses how participants managed to integrate knowledge, both tacit and explicit knowledge, through iterations of individual action and collective interaction. From Lindkvist's et al. (2011) study one notes how knowledge integration and knowledge creation can be conceptualized in terms of three collaborative processes. Hargadon and Beckhy (2006) examine collective creativity and explain four underlying collaborative activities (help giving, help seeking, reframing, reinforcing). Kellogg et al. (2006) and Matusov (1996) stress the emergent, dynamic and creative aspects of knowledge integration and discuss coordinating activities and process features by referring to concepts such as 'trading zone', 'multiparty coordinated action', 'emerging collage', and 'progression in mosaic fashion'. Teasley (1997) relates effective problem-solving to a certain conversation style called 'transactive reasoning'. Nonaka (1994) and Tsoukas (2009) argue that new knowledge is created in face-to-face dialogues. Both authors discuss how new concepts emerge. Nonaka (2004) highlights the use of metaphors, and Tsoukas (2009) stresses the importance of personal attitude and relational engagement. These ideas on how knowledge integration occurs, along with the different knowledge integration mechanisms presented in section 2.2.2.1, are summarized in the table below:

Author	**Knowledge Integration occurs through:**	**Type, keyword, characteristic**
Andersson and Berggren (2011)	Agents	Innovators, knowledge integration-agents
Iansiti (1995)	Agents	T-shape skill pattern
Dahlander and O'Mahoney (2011)	Agents	Lateral advancement, individual role and coordination work
Grant (1996a,b)	Mechanisms	Sequencing, rules and directives, organizational routines, group problem-solving
Schmickl and Kieser (2008)	Mechanisms	Transactive memory, Modularization, Prototypes
Okhuysen and Eisenhardt (2002)	Mechanisms	Managing time, questioning others, (information sharing)
Söderlund (2010), Lindkvist et al. (1998)	Mechanisms	Time management, milestones, deadlines
Faraj and Sproull (2000)	Mechanisms	Knowing Expertise Location, Recognizing the Need for Expertise, Bringing Expertise to Bear, Software development methods, milestones, review meetings.
Tiwana and McLean (2005)	Mechanisms	Relational Capital, Absorptive Capacity
Majchrzak et al. (2012)	Activities	Voicing fragments, Co-creating a scaffold, Dialoguing around the scaffold, Moving the scaffold aside, Sustaining engagement
Majchrzak et al. (2007)	Activities	Running narratives
Faraj and Xiao (2006)	Activities	Expertise Coordination Practices and Dialogic Coordination Practices
Okhuysen and Bechky (2009)	Activities	Integrating conditions, coordination as an ongoing accomplishment
Söderlund (2010)	Activities	Knowledge Entrainment, Synchronization, Pacing, Orchestration
Enberg et al. (2006)	Activities	Acting and Interacting
Lindkvist et al. (2011)	Activities	Collaborative processes: Pre-selection, variation, selective retention
Hargadon and Bechky (2006)	Activities	Help seeking, help giving, reframing, reinforcing
Kellogg et al. (2006)	Activities	Trading zone, display, representation, assembly, Emerging collage
Matusov (1996)	Activities	Progression in mosaic fashion

Pinto & Pinto (1990); Daft & Lengel (1986); Wheelwright and Clark (1992); Li et. al. (2007); Brown and Eisenhardt (1995); Sapsed & Salter (2004); Hedlund (1994); Grant (1996a,b)	Interpersonal communication	Unregulated, informal, rich, dense, frequent, face-to-face interaction and group problem-solving,
Nonaka (1994)	Interpersonal communication	Creative dialogue
Tsoukas (2009)	Interpersonal communication	Productive dialogue, relational engagement
Teasley (1997); Berkowitz (1993)	Interpersonal communication	Transactive reasoning
Okhuysen and Eisenhardt (2002)	Interpersonal communication	Other-directed communication

Table 1: KI Agents, Mechanisms, Activities and Communication

Although these are all informative and attractive accounts of knowledge integration and creation, they are generally not related to an analysis of a specific development process from start to end, and cannot therefore easily explain how mechanisms, underlying processes, and activities of KI change are connected to specific situations. Two exceptions to this general trend are the studies that were conducted by Faraj and Xiao (2006) and Majchrzak et al. (2012). Such mechanisms, processes, and activities might benefit from being framed in space and time to further understand what triggers change during a particular development process. Adjusting processes and mechanisms in accordance with unanticipated problems and "everyday dynamics" is important to ensure that the overall development and knowledge integration process does not break down. It is also necessary to operationalize knowledge integration and make the dynamics of knowledge integration observable in "real life". The following sections discuss where knowledge is integrated and how to capture the dynamics of the knowledge integration processes.

2.3 Where is knowledge integrated?

In the first part in this section, 2.3.1 the cross-functional project is characterized and it is argued that it is a common setting for knowledge integration. In the second part, 2.3.2, a problem-solving approach to studying knowledge integration in development projects is suggested.

2.3.1 Knowledge integration and cross-functional development projects

If researchers and practitioners look upon knowledge as something that resides within individuals and is, to a great extent, tacit (and thereby difficult to transfer), it is not difficult to understand the past decades' "vogue for team-based structures" (Grant 1996a p.118) and cross-functional projects. This kind of organizational design entails that specialists from different organizational units are directly involved in decision-making and participate in the operative development work. Direct involvement is assumed to be crucial for successful integration, since managers and traditional hierarchy structures and processes cannot easily access and make use of the specialists' inner knowledge and know-how (Grant 1996a).

Nonaka (1994) underlines the importance of cross-functional self-organizing teams to a firm's knowledge creation process. The team is the setting or the "field" where individuals meet and interact to create new concepts. He argues that deep commitment, mutual trust, and continuous dialogue between members are essential aspects of effective team-work. Huang and Newell (2003) argue that cross-functional teams are employed when creativity is important to solve the task at hand and if participants are expected to generate new ideas and solutions, as is the case in, for example, new product and systems development. Tiwana and McLean (2005 p. 14) assert that information systems development is a creative endeavour involving the "generation and evaluation of new ideas, designs, solutions, and artifacts" which requires the participation of many individuals and the integration of different kinds of expertise at the project level. Huang and Newell (2003) state that cross-functional teams are also important for managing strategic change and planning within firms since such changes (for instance, new technological solutions or new business process innovations) and decisions often affect many different groups and units in the organization whose different skills, perspectives, and interests must be applied, resolved, and incorporated into the new technology and processes. Hobday (2000) and Söderlund and Tell (2011) also argue that the project organization is an important setting where specialized knowledge that has been developed locally in different units is integrated so as to create complex products and systems. As Tell (2011 p. 36) states "much knowledge integration and innovation goes on in projects". Söderlund (2010) argues that multifunctional development projects and product teams can be seen as knowledge integration mechanisms, not least because of its 'macro pacing' function.

Although the importance of cross-functional teams and projects is widely recognized, Huang and Newell (2003) argue that the general understanding of knowledge integration processes within cross-functional projects is limited and needs to be further explored. As mentioned earlier, Huang and Newell (2003 p. 168) define cross-functional project teams as "groups which have members with highly differentiated knowledge and a mission that can only be fulfilled through integrating the differentiated knowledge both internally within the group and externally with the various stakeholder groups impacted by the group's work." This describes the empirical setting that was studies in this thesis quite well but it neglects to include one crucial element that is central to understanding projects as theoretical phenomenon and my specific case project – the temporary aspect of the organization (Sydow et al. 2004; Lundin and Söderholm 1995). Sydow et al. (2004 p. 1480) suggest the following definition of projects: "projects as temporary systems refer to groups comprising a mix of specialist competences, which have to achieve a certain goal or carry out a specific task within limits set as to costs and time." They emphasize the fact that "such a view is informative of the transient and multidisciplinary nature of projects." (Sydow et al. 2004 p. 1480)

The project that was studied for this thesis corresponds well with this definition since it was a temporary and time-limited complex undertaking (requiring different specialist competencies) around which a cross-functional team was formed. When the mission was completed, the whole project organization dissolved. The project and the task was also unique for the organization which normally also constitutes a central characteristic of development projects (Söderlund 2004; Engwall and Westling 2004). The project members also talked about their work situation and task in terms of "project assignment, subprojects, project managers, project management, project documents, project members, project task or goal, the problem that the project should solve, project plans, project activities".

For the purposes of the present study, some important features were that the project task was complex and had not been performed in the same way before, and that the task required different competencies. The complexity of the task and the organization (e.g. technical aspects, number of team members) and the limited time span must also be suitable for this kind of research. However, as Lundin and Söderholm (1995 p. 439) state, time, task, team, and transition (the transformation or change that the project should accomplish) "define the action arena, but do not explain the actions performed in that arena." Söderlund (2004 p.185) argues that “projects are important and interesting phenomena from which it is possible to build strong and interesting theories in order to increase our knowledge of certain parts of social life”. This means that even if the project conditions existed, the most critical thing was whether there was opportunity to capture knowledge integration processes within this setting. In this context, we will examine knowledge integration processes by studying the actions that are performed by the project members and managers over time. Since our interest lies in understanding the dynamics of knowledge integration (how knowledge integration unfolds over time from start to finish in a particular project) I chose to pay extra attention to unanticipated problems and challenges that may require *changes* in current activities and the use of integration mechanisms. Since both practitioners and scholars have reported on the frequency of problems (related to collaboration, technology, external factors, and different kinds of uncertainty and ambiguity) in projects that need to be solved quickly, a development project appeared to be an appropriate setting. The problem-solving character of development projects will be elaborated on in more detail in the next section with reference to the literature on problem-solving from the domains of cognitive science and behavioural economics, which complements scholarly work from the fields of organization theory and new product development.

2.3.2 A problem-solving approach to study KI processes in projects

The argument for using a problem-solving approach to study KI processes in projects stems from the idea that problems and the collective search for solutions are expected to offer an insight into the everyday dynamics of the project and how knowledge integration processes continuously adapt to meet new challenges and situation. Problems might “disturb” the current order and ways of working and prevailing knowledge integration mechanisms and activities may be ineffective when circumstances change.

The outline of this section is as follows. First is a short background discussion of why problems are so frequent in development projects, why it is difficult and even counterproductive to make detailed plans and goal specifications, and how project members typically deal with these issues. Then a short description is given of problem-solving processes, which is based on the literature on organization theory, project management, and new product development. After this discussion, I provide some further discussion of problem-solving processes and strategies which is based on previous work done by problem-solving researchers in cognitive science and behavioral economics. Finally, I suggest how we can understand problem-solving in relation to the interest and study of dynamic knowledge integration in development projects.

Background to problem-solving in development settings: Various researchers have pointed at the ambiguity and uncertainty that characterize development settings and have argued that project workers often must deal with unexpected problems throughout the development process. Even though project managers develop plans, schedules, and milestones, and attempt to control the project by using different project management models (PMMs) that are “tied to a variety of technocratic planning, execution and reporting tools to ensure that projects are run rationally according to set budgets, goals and time schedules”, (Räisänen and Linde 2004 p.103), there is still much ambiguity and uncertainty that cannot be taken away at the beginning of the project process (Lindahl 2003; Christensen and Kreiner 1991). Project members do not know exactly what will happen during the course of the project; what problems, possibilities, or solutions will emerge and be effective. Technical difficulties and challenges also give rise to ambiguous problem situations. Development undertakings carried out in cross-functional projects may even be portrayed as goal-driven problem-solving processes (Lindkvist 2008). The modern market's inherent complexity and uncertainty has made it difficult to plan and specify project goals and outcomes in detail – instead, it has shown to easily result in bad market adaptation and project failures. How do project management and members deal with this?

Engwall (2002 p.273) asserts that the goal should be seen as an "abstract vision that provides a direction" to the project members. It is a hypothetical conjecture on what position or result is desired on a certain day in the future. He found that the project goal and the plans associated with this goal are based on a process of "*abstract thinking*" and that project execution is a process where pre-set theories co-act with the acquisition of practical knowledge created during the realization of the different activities. Importantly, instead of planning exactly, project members and managers must improvise (Lindahl 2003) and process problems and solutions iteratively and learn along the way from their experience. Cross and Sproull (2004) argue that in knowledge-intensive work such as software development, problems of different kinds turn up which, due to compressed project time frames, must be solved "right here, right now" to avoid losing time and speed. Problem-solving in these settings aims at creating "actionable knowledge", which, in their view, is pragmatic goal-directed knowledge that "leads to immediate progress on a current assignment or project" (Cross and Sproull 2004 p. 446). Many scholars have noticed that trial-and-error, which involves brainstorming, imagination, guessing, and experimenting, constitutes a prominent feature of development processes (Lindkvist 2008, Nickersson and Zenger 2004, Thomke and Fujimoto 2000). This kind of problem-solving activity is explained in more detail below.

Problem-solving processes in new product development: Iansiti (1995 p. 523) argues that "problem-solving activities are a fundamental engine in technological evolution. They drive the evaluation of new ideas and the generation of knowledge". The author further asserts that "effectiveness in product development is linked to the problem-solving behaviour of individuals in an organization". Nickerson and Zenger (2004 p. 618-619) also stress the importance of effective problem-solving and state that if a firm is able to create unique knowledge or an exclusive capability "through any other manner than luck", then it must detect and define an important problem and carry out an efficient solution search process.

Thomke (1998 p. 743) discusses problem-solving experimentation in terms of "trial, failure, learning, correction and retrials". In a similar vein, some researchers discuss "problem-solving cycles" in which a steady stream of sub-problems are dealt with and potential solutions tried until the whole product is developed (Brown and Eisenhardt 1995, Sheremata 2000). Thomke and Fujimoto (2000) show that, as a project progresses, the iterations or cycles include models such as computer simulations, prototypes, or pilot vehicles of increasing completeness. Hippel and Tyre (1995 p. 2) note that project members in their study first generated different alternative solutions, then tested them against "a whole array of requirements and constraints". The test outcomes were then later used to elaborate and refine earlier solutions.

Researchers have emphasized the importance of careful problem formulation and representation because, as Schön (1983) and Westling (2002) reason, in uncertain and puzzling situations, as, for example, in product development projects, problems do not present themselves as given. Westling (2002 p. 3) argues that solving complex and ambiguous problems in product development projects involves "the identification, framing and defining of problems that convert them into solvable problems." Cross and Sproull (2004 p. 446) argue that problem-solving in complex development work involves “defining relevant dimensions of a problem space, crafting a solution that is both feasible and appropriate for the social context where it will be introduced, and convincing others of the correctness of a proposed course of action.” This introduction to problem setting and problem-solving now leads us to consider problem-solving processes in the world of cognitive science and behavioural economics.

Problem-solving processes in cognitive science and behavioral economics: In the research domains of cognitive science and behavioral economics, I have found three, different but related, sequential sub-processes or activities that constitute problem-solving. The first activity is concerned with the importance of finding or creating a definition of the problem to be solved. The next activity involves the actual search for a solution. The last activity deals with the evaluation of the solution and of one’s own norms and hypotheses. I will explain each activity in more detail below. The actual search for a solution receives the most attention since it includes a concept, ‘heuristics’, which seems to be of value to understanding the field of organization theory in general (Bingham and Eisenhardt 2011), and my empirical case project in particular.

Identifying an interesting problem and representing it in a beneficial way is a complicated activity. Dunbar (1997) states that a problem can be represented in many different ways and that some representations are better suited for finding a valuable solution than others. Problem-solving researchers declare that defining the problem involves setting the problem space (Newell and Simon 1972, Dunbar 1997) or solution landscape and its interacting parameters (Gavetti and Levinthal 2000). It frames the course of subsequent action and affects the decisions that people will make (Kahneman and Tversky 1984). The solution landscape becomes a “pre-choice” of what is regarded as relevant in the following phase. Schön (1983) argues similarly that setting a problem means to frame the context and choose what to prioritise and regard as important elements of a situation. However, Kaplan and Simon (1990 p.376) argue that individuals “do not initially choose deliberately among problem representations, but almost always adopt the representation suggested by the verbal problem statement.” The researchers assert that problem-solvers do not easily change their initial representations and argue that there is a need for more studies on how individuals initiate new problem representations and formulations.

The second phase of the problem-solving process concerns the actual search for a solution. Dunbar (1997 p. 5-6) argues that “one of the most important aspects of problem solving becomes one of searching for a path through the problem space that will lead to the goal state”. Problem-solving researchers (e.g. Kaplan and Simon 1990, Dunbar 1997) have studied how problem-solvers create and use different rules of thumb, so called *heuristics*, when solving complex problems and dealing with problematic situations. Dunbar (1997 p. 5-6) argues that employing heuristics can be seen as a crucial strategy when searching large problem spaces and solving complex problems, because problem-solvers cannot keep in mind a whole complex space with all its details; “often, problem solvers will only have a small set of states of the problem space represented at any one point in time” and need therefore heuristic guidance that “allow them to move forward through the space.” Notwithstanding this, while rules of thumb may lead to the right solution, it does not guarantee the correct solution, as Dunbar (1997 p. 5-6) argues:

> In problem solving research, a heuristic is a rule of thumb that will generally get one at the correct solution, but does not guarantee

> the correct solution. An example of a heuristic might be that "If I start playing tic-tac-toe by putting an X in the middle square, I will win. This heuristic does not always work; sometimes I lose even with this strategy!

Eisenhardt et al. (2010) have also touched upon a similar topic recently and assert that using heuristics in problem-solving is particularly valuable when time is scarce. This is because heuristics provide shortcuts in the problem-solving process. Heuristics are easy and quick to use, and when compared to routines, are flexible and temporary and suit ill-defined problems. Ill-defined problems and time pressure is something which development project members often face. Eisenhardt et al. (2010 p. 1266) argue:

> Heuristics are rules of thumb that provide shortcuts in problem solving […] Heuristics emerge as individuals adjust to problem-solving situations in which there is limited time and information […] Because heuristics are easy for organizational members to remember and are quick to use, they provide efficient guidance for some actions, but just as important, they also leave room for flexible adjustment in real time of other actions. Heuristics are thus distinct from routines that provide detailed and automatic guidance for well-specified problems and so favour efficiency.

Various heuristics with different levels of complexity have been identified in cognitive research and behavioral economics. I will give some examples of different heuristics before moving on to the final problem-solving sub-process, *evaluation*.

Heuristics are general problem-solving strategies - not specifically related to "real-life" work in development projects, but may inspire one to think of how project workers solve complex problems. For example, "simple search" is one heuristic that involves randomly choosing what to try and where to go when solving a complex problem. This strategy is most often used when people have no clue what will take them to the goal (Dunbar 1997 p. 6). A somewhat more complex strategy is "to move to the state that looks most like the goal state" Dunbar (1997 p. 6) argues. This is called "hill climbing" (Dunbar 1997). It might be an effective strategy if the problem solvers can see more than one step ahead. But it can also still lead them off track if the distance to the goal is very great and includes many steps. One or two steps ahead that might seem to lead to the goal may still be too little to judge whether or not it is a path that will actually bring the problem solver to the goal.

Another heuristics is 'means-ends analysis', which involves an analysis of a current state of affairs in comparison to the desired goal state and an identification of potential hindrances that block the way towards the goal. A sub–goal, to eliminate the first hindrance, is defined and the problem solver then analyzes the situation again in the same way, and "decomposes the difference between the current state and the goal state into another sub problem and sets a goal of solving that problem" (Dunbar 1997 p. 6). This process continues until the problem is solved. Hill climbing, trial-and-error, and means-end analysis are iterative search heuristics.

Yet another form of search strategy or heuristic is 'analogical reasoning' which may be a powerful and efficient strategy to employ if the search space is very large (Dunbar 1997). "If the problem solver has solved a similar problem in the past, she or he can go directly to the solution by mapping the solution to the old problem onto the current problem." (Dunbar 1997 p.7) In this way, the problem solver can "jump from one part of a problem space to another bypassing many of the intermediate states" (Dunbar 1997 p.7).

Lyles and Mitroff (1980) argue that problem solvers tend to use simple problem-solving techniques aimed at well-defined problems even for ill-defined or complex problems. Similarly, Kahneman (2003) discusses 'attribute substitution heuristic' as a problem-solving method, which entails that a problem solver creates an answer to a simpler problem or question than the one that was actually raised. Problem solvers make things more straightforward than they really are, without knowing it. "A judgment is said to be mediated by a heuristic when the individual assesses a specified target attribute of a judgment object by substituting a related heuristic attribute that comes more readily to mind […] The cognitive illusions that are produced by attribute substitution have the same character: An impression of one attribute is mapped onto the scale of another, and the judge is normally unaware of the substitution." (Kahneman 2003 p. 707). This means that "people who are confronted with a difficult question sometimes answer an easier one instead". (Kahneman 2003 p. 707).

After a solution and conclusion have been reached, the steps of the process should be reconsidered to analyse what was "helpful, harmful, or merely useless" so as to learn how similar problems may be attacked in the future (Dewey 1910/2007 p. 112-113). This takes us to the final step in the problem-solving process, *evaluation of solutions and hypotheses*. In problem-solving research, it has been noted that individuals do not easily give up their ideas and hypotheses, not even in the face of disconfirming test results. Klahr and Dunbar (1988) argue in their study on how people propose and reformulate hypotheses and generate experiments to test their hypotheses that people reveal strong confirmation bias; they see and choose what confirms their hypothesis, instead of looking at aspects that may disconfirm the hypothesis.

Klahr and Dunbar (1988) also found a number of reasons why problem solvers frequently hold on to their current hypotheses even in the face of negative information. One reason is that they tend to distrust the feedback that is produced in the tests or experiments and classify "disconfirming instances as erroneous trials" (Klahr and Dunbar 1988 p.43) (e.g. fallible test device, test data). Moreover, the subjects' prior knowledge can result in hypotheses that have high *a priori* strength and therefore need much disconfirming evidence to be rejected. Another reason why people tend to maintain their initial hypotheses is that they cannot think of alternative hypotheses. If the subject is able to replace disconfirmed hypotheses with "nothing", then problem solvers can more easily drop rejected hypotheses. Klahr and Dunbar (1988) argue that if there is a low probability that a certain hypothesis will be confirmed, positive test results are more valuable than if there had been a high probability that the hypothesis would be confirmed. Likewise, when the probability of confirming a hypothesis is high, negative instances provide useful and interesting information. Consequently "the appropriateness of the strategy depends on the distribution of positive and negative instances" according to Klahr and Dunbar (1988 p. 42).

In pragmatic thinking, new solutions and new knowledge and concepts are evaluated in relation to their practical usefulness or purposefulness (Cook and Brown 1999). Since some researchers and philosophers argue that all knowledge is uncertain, evaluation of test results is often a question of deciding what is better and preferable compared to something else (Lindkvist et al. 2011). Project members assess whether the outcome of the process is better than the preceding solution, and whether it then should replace the current solution. Prevailing norms and assumptions should be critically examined and systematically questioned, not only to detect and correct small local errors, but also to form better strategies for future development, which is important to prevent dysfunctional routines or habits that impede innovation.

Problem solving and dynamic knowledge integration: Now we have reached the last part in this section that considers problem solving as a tactic to capture the dynamics and study knowledge integration processes in development projects. I will here explain what I intend to do with the information just presented around problem solving and how it facilitates the research on knowledge integration processes.

First, we can draw parallels between problem-solving processes involving problem formulation, the search for a solution and evaluation of results and hypotheses, and the overall ISD project process and knowledge integration process. Tiwana and McLean (2005 p. 16) describe an information systems development process and report that "[t]he systems development life cycle involves translating an abstract business idea into project requirements", which can be seen as a problem identification process. The requirements "are then used to create project concepts and systems specifications, and eventually the functionality and features in the software code" (ibid.), which can be likened to a solution search process. The authors argue that "there is rarely 'one right design' for an ISD problem, because there is often more than one possible solution to the same end" (ibid.). They state further that "[s]uccessful ISD thus depends on a team's developing a preliminary idea beyond its embryonic state by drawing on several interdependent bases of expertise." (ibid.) "Depending on how creative the process is, a team might come up with many possible solutions to the same problem." (ibid.) This, in essence, constitutes knowledge integration and creation through solution search processes. After a beneficial solution has been accomplished the next step is about convincing others about the correctness and appropriateness of the solution and determining whether the new solution is better than existing one (Cross and Sproull 2004, Lindkvist et al. 2011). This resembles the evaluation part in a problem-solving activity.

The problem-solving approach thus provides us with a useful starting point in understanding the different project sequences from the beginning to the end, at a general level. We can, however, also expect to see a good number of sub-problems, trail-and-error episodes, and parallel work – the problem-solving approach *may help in framing* the analysis how underlying knowledge integration processes, activities, and mechanisms unfold and change in a development project over time. However, various researchers have shown and cautioned their readers that traditional project life-cycle sequences or linear models stemming from engineering and project management schools are not useful ways to understand project dynamics and how projects develop over time (Söderlund 2004, Engwall and Westling 2004, Lundin and Söderholm 1995, Gersick 1988). This should be kept in mind when interpreting the case project.

The details of the problem-solving activities, especially the *heuristics concept*, can help bring new understanding of how problems are solved in time-pressured situations, and how such problem-solving strategies can keep the knowledge integration moving forward. I will relate problem-solving activities to the dynamics of the knowledge integration process. For example, one can imagine that, as long as no difficult problems turn up, things will proceed according to plan. However, if problems occur that cannot easily be solved, the knowledge integration process might be impeded. When problems are being solved one might wonder whether changes have been made in the knowledge integrating activities and mechanisms too so as to support the new working order or whether it might be the other way around; did changes in the knowledge integration processes and mechanisms facilitate problem-solving? What problems and challenges do the project members face in different development phases related to joint problem-solving and integration of knowledge? Are different problem-solving and knowledge integration mechanisms in use in different situations and stages? What makes them change; what drives the process of change? The general question is *how do individuals solve novel problems together and integrate their knowledge over time in a certain development process*? In summary, problem-solving processes and activities are supposed to facilitate the identification of dynamic knowledge integration activities and underlying processes and mechanisms over time in a specific cross-functional development project.

A final important issue remains to be discussed in this chapter. It concerns the differences in knowledge and interpretations that the different specialized project members bring to the task. Up to this point, the main problem has been on understanding how specialized knowledge can be integrated. However, as Grant (1996a) and many researchers with him have argued, this is not a simple task, especially not since a large part of the knowledge that is to be integrated is tacit and difficult to articulate. Different specialists may have difficulties in understanding each other and may also have divergent opinions and interests that impede integration and fast development. The specialists' knowledge and ideas must, however, be used and integrated in the problem-solving process appropriately over time. Thus it might be necessary to encourage the appearance of individual knowledge in some situations, whereas, in other situations, it might be more appropriate to limit the emergence of divergent ideas and solution alternatives. This will be explained in the following section.

3. Promoting and constraining diversity appropriately over time

The third part of this chapter treats the dilemma of on the one hand use and exploit the creative power of diversity in order to achieve innovative solutions, which most probably is time-consuming, and on the other hand avoid diversity-related friction and conflicts that may hinder decision making, development and rapid progress. This problem entails finding ways to stimulate the appearance of different people's unique knowledge while at the same time inventing measures to hold knowledge differences back in order to progress as fast as possible. One challenge is thus for project members to unleash and apply their ideas, unique knowledge, and expertise at the "right" time during a development process (Larson 2007) and find appropriate integration mechanisms and problem-solving strategies that enable this.

The next section, 3.1, discusses the problem of making unique knowledge accessible in development and problem-solving processes. Then some measures that may encourage the appearance and utilization of unique expertise are presented. In 3.2, the opposite question is discussed; how can the space for expressing diversity be constrained by the use of different knowledge integration mechanisms? Finally, the challenge to balance the encouragement and restraint of diversity appropriately over time in a development process is discussed.

Before we proceed, the term *diversity* will be explained and defined, as well as the idea of widening and narrowing the space for expressing difference. *Diversity* refers to the cross-functional project members' variety of specialized knowledge and functional background. As Majchrzak et al. (2012 p. 966) do, I use the term *cross-functional* to refer to the "different perspectives that team members bring with them" […] It does not refer to their demographic diversity (e.g., gender, race)." Bunderson and Sutcliffe (2002 p. 875) use the concept "functional diversity" which is associated with tenure background and "differences of opinion and perspective". Cronin and Weingart (2007 p. 761) also use the concept "functional diversity" and assert that knowledge differences can easily result in "*representational gaps* – inconsistencies between individuals' definitions of the team's problem" and joint problem solving difficulties. Rico et al. (2008 p.171) discuss "[k]nowledge diversity" and suggest that it "refers to the distribution of knowledge relevant to the purpose or task of a team among its members" which resembles with the use of the concept in this thesis. In the literature on creativity, knowledge integration, knowledge creation, and new product development, researchers discuss how cross-functional members' knowledge differences may impede innovation and development since the different specialized knowledge bases required to create new complex products and services also hinder innovation since knowledge differences make communication and cross-functional coordination difficult. In this study, diversity is thus only connected with knowledge and differences in perspective that are relevant for the development task and work among team members. Again, it should be noted that diversity is not associated with demographic factors such as age, sex, or race. The term *diversity* could, for instance, be exchanged with *knowledge differences* or *expertise heterogeneity* or *knowledge differentiation*. These interrelated concepts indicate that diversity is about knowledge that is related to work contexts, problems and tasks.

When I discuss how to manage diversity in terms of promoting, encouraging, utilizing, constraining, or restricting diversity, I mean that the possibilities for individual actors to express viewpoints or perspectives and develop and apply own ideas or personal perspectives are influenced or steered in some direction. The "room" or "space" for expressing difference or divergent action can be either enlarged or narrowed. Managing diversity may also imply that the total variety of specialist competences in a team can be increased by the employment of new people with other skills or reduced by discarding particular team members with unique (in relation to the other team members') knowledge.

3.1 Promoting and utilizing diversity

Framing and solving complex problems typically involves knowledge that is distributed among different specialists (Nickerson and Zenger 2004, Sheremata 2000, Tiwana and McLean 2005; Li et al. 2007; Iansiti 1995). As discussed by Dunbar (1997), diversified groups may generate more alternatives and representations of a problem than a single individual. Complex problem-solving should be seen as a matter of 'distributed reasoning' among diverse individuals who are experts in different knowledge domains and therefore naturally focus on different parts of the problem. Dunbar (1997 p. 13) notes that "[i]f all members of a group are from the same background, they tend to represent the problem in the same way. If their representation is incorrect they fail to solve the problem." Tiwana and McLean (2005 p. 19) argue that creativity in solutions depends on the project members' capability to integrate and find "novel associations and linkages among the diverse ideas, perspectives, and domain expertise that individual team members hold." If the project manages to integrate individually-held expertise, it will enhance team creativity "because it leads team members to access, explore, and use diverse information from related knowledge domains associated with the project" (Tiwana and McLean 2005 p. 19). Iansiti (1995) argues that effective problem framing and problem solution and knowledge integration is associated with information searches in many different disciplinary knowledge bases, including searches in previously unrelated knowledge domains. A major challenge in the development process is thus to encourage people to use their unique knowledge and insights, and generate a wide range of alternatives and ideas and a varied supply of potential solutions. This is discussed below. I will here refer to some scholarly work in *social psychology* and *small group research*.

3.1.1 Unique knowledge difficulties

Unique knowledge does not easily surface in group interaction and conversation, and if it does, it runs the risk of being ignored or turned down, something that is sometimes referred to as the 'common knowledge effect' (Gigone and Hastie 1993). These authors found that groups often did not actually take unshared information into consideration in decision-making. Instead, judgement was more often based on previously held opinions and the information that most members already were aware of. People tended to discuss what they shared, instead of unshared information. Hence, as commented on by Stasser (1999), the impact of unique knowledge in collective choice is lower than we might think in actual situations.

Unshared information is less likely to be repeated once it has appeared, when compared to shared information. Larson et al. (1994) shows that, in untrained groups, shared information was mentioned earlier in the conversation process than unshared information. Factors that may shorten discussions, such as early consensus and time pressure, may thus prevent group members from reaching the phase or point in the discussion where unshared information starts to appear and have a significant impact.

Stasser (1999) also discusses different reasons for why unique information is less likely to be shared than common knowledge. One explanation is that, as unique information is displayed, it may be the first time people hear it. When shared information is mentioned, it is, at least, the second time people are exposed to it. Repeated information tends to have an advantage over new information in terms of its influence on decision making. Another explanation why unique and "new" knowledge does not easily surface, which Stasser compared to Janis' notion of "groupthink", is that new information might be seen as threatening or disturbing, or as a hindrance to task resolution, consensus, or agreement.

In summary, Stasser et al. (1989), Stasser (1999), Larson et al. (1994) and Gigone and Hastie (1993) show that there are many obstacles to making the most of unique knowledge in problem-solving processes. Fortunately, there are a number of ways to facilitate the appearance and use of unique knowledge.

3.1.2 Encouraging the appearance of unique knowledge

As is discussed by Stasser et al. (1995), group members are more likely to mention unique or unshared information if they know that they are experts relative to others in a certain area or in relation to a specific task. New information is often treated with some doubt and given less attention if it comes from a non-expert when compared to a speaker who is an expert. An expert does not seek or need the same amount of confirmation for new information as a novice does or as those who believe they are non-experts do (Stasser 1999). It is also essential that all participants mutually recognize and accept each other's domain knowledge (Stasser et al. 1995). The chances for unshared information to be mentioned in conversation will also increase if the unshared information that is enunciated by one speaker becomes socially validated by other participants. This is especially so if the interlocutor is uncertain, does not know whether he or she is an expert, or is not generally viewed as an expert by the other members.

Displaying a great deal of unique information may not, however, result in that person being seen as an expert by collaborators (Kameda et al. 1997). What counts, according to Kameda et al. (1997), is a person's position in the cognitive network. In order to possess a central position and be an influential person, one should share as much knowledge as possible with other team members so that one can validate other people's contribution and also express oneself so that one's own contributions are validated. Cognitive centrality, as Kameda et al. (1997) call it, signifies expertise. This is perhaps contrary to what one may initially assume. A person who knows a little bit of everything should rather be viewed a generalist and not an expert. Kameda's et al. (1997 p.306) study shows, however, that "[e]ven when they were in the preference minorities, cognitively central members exerted more influences on group decisions, guiding consensus outcomes toward their preference to a larger extent than peripheral members."

Stasser (1999) also suggests some more practical tricks-of-the-trade that may help the sharing and recognition of unique information. For example, if team members are asked to rank different alternatives, then instead of merely choosing the best alternative, more unique information is exchanged. If group members have fewer facts to consider, they do not overlook unshared information as much as when they are overloaded by information (Stasser 1989). In addition, structuring the discussion by asking the participants to first discuss different alternatives without a specific preference and evaluation, and then, after some time, asking them to judge, choose and make a decision, may result in more information being mentioned when compared to unstructured discussions.

On the whole, this means that groups are often more inclined to focus on coordinating and integrating their common knowledge than their dispersed unique knowledge but some ideas exist that are informative in a general sense of how one may facilitate the occurrence and use of unique information.

3.2 Containing and constraining diversity

Diversity, associated to functional background and education, has been related to effective decision-making and action. However, when project members define the problem and task very differently and have widely divergent opinions on the best way of approaching a task they will probably experience communication difficulties and have problems with deciding what to do. The members will then not easily see or appreciate each other's perspectives either (Kurtzberg and Amabile 2001). Kurtzberg and Amabile (2001 p. 290) discuss how diversity in knowledge, on the one hand, can fuel and spur the creative process by providing diverse perspectives and ideas. On the other hand, it can also hinder group process by "creating such a divergence of ideas that detrimental conflict can result" (ibid.) and the project will then run the risk of stagnating. Members from different departments "not only know different things, but also know things differently" (Dougherty 1992 p.187). Thus, while diversity, in terms of disparate knowledge bases among project members, is certainly important in achieving creative solutions to complex development tasks, it also easily causes disruptive interpersonal conflicts, unproductive discussions, indecisiveness, and interdisciplinary quarrels.

Whereas diverse skills and multiple viewpoints should be recognized, fostered, and integrated in order to achieve creative solutions, it may also occasionally be necessary to constrain people's "space" for expressing opinions, knowledge, and creative thinking, in order to swiftly solve problems and move forward. Integrating disparate knowledge, ideas and perspectives and, at the same time, moving forward in a time-efficient manner, can be a challenging task. Projects are generally time-limited endeavours, which, in the empirical case used in this thesis, involve compressed time frames and a great deal of pressure to work intensively and solve problems quickly. Hence, reaching the project's goal in time will not only be a matter of promoting and using prevailing diversity, but also of occasionally constraining the application of this diversity.

The following section will discuss Kellogg's et al. (2006) finding that the knowledge integration mechanism "standardizing media" has a reducing effect on "creative abrasion" (Leonard-Barton 1995) across different occupational groups. I will also discuss how knowledge integration mechanisms and interventions such as deadlines and time restrictions, rules, routines, directives, and sequencing may also have delimiting effect on the members' possibilities to display diversity. By referring to *creativity research* I will suggest how evolving interaction patterns and structures may constrain action possibilities and individual divergence.

Kellogg et al. (2006 p. 23) found that in heterarchical organizations (Hedlund 1994) that "rely more on horizontal than vertical relations, utilize multiple, shifting centres of expertise and accountability, and work through rapid assignments, temporary agreements, and creative misunderstandings rather than standardized routines and stable commitments", the space and moments when differences blend were constrained by the standardization of how individuals communicated and used media. By the use of "internal and external networks, electronic mail system, online calendaring tool, and project management system" the project members could focus on fast coordination and operation "through shared *forms* of communication rather than shared *content"* (Kellogg et al. 2006 p. 30). However, while such standardization allows for local differences in disparate communities and enables coordination without cross-learning, knowledge transfer or shared agreements, it also restrains difference. The project members had to decrease the amount of detail that they shared and filter the information before displaying reports, messages, status updates, and presentations to other groups so that the information would fit with a standardized communication format. Local details were excluded, in order to make the documents and information legible for others. The project "genre" which standardized the appearance by using uniform layouts and shape of PowerPoint presentations, documents, and emails made it difficult to sort among facts and information and distinguish between messages and details. "The flattening of content in e-mails and presentations, while facilitating project-wide legibility, inhibited the texture and richness that is often the stuff of creative friction, thus potentially stifling innovation." (Kellogg et al. 2006 p. 40) In this way, homogenizing a communication form and standardizing media may result in a decrease in the moments where knowledge differences across various functions meet and amalgamate.

Okhuysen and Eisenhardt (2002) discuss "managing time" as a mechanism for integrating knowledge, as mentioned in section 2.2.2.1. They found that if members are aware that the time available to solve a particular task was scarce, then they integrated their different perspectives and knowledge more efficiently. Lindkvist et al. (1998) studied new product development and found that short deadlines were of great importance to knowledge integration since they resulted in that project members must coordinate actions quickly and create 'good enough' solutions. However, the creation of good enough solutions may involve that project members' unique knowledge and expertise are not being fully utilized. Tyre et al. (2002) also discuss how temporal shifts, that is to say, a change in the normal work rhythm, have a coordinating effect in organizations, since organizational members are engaged in new forms of interaction that brings diverse individuals together. These individuals then focus more on group problems than individual problems. Gersick (1989) has studied the role of deadlines and time pacing and has shown that project members who are subjected to time pressure hurry themselves, call attention to the time table, and by the midway point of the project, assess that work that was completed against remaining time and work. To move forward, project members summarized the process and task accomplishment, and, at that time, decided to stop producing ideas and closed off alternatives. The members prevented further discussion and tried to narrow down the work to make decisions on what to focus on and prioritize. One can thus see that the time pressure indirectly constrained people's room to be creative, at least temporarily. Furthermore, Okhuysen and Eisenhardt (2002) note in their study that time pressure can also result in someone who starts to dictate and act as a leader occasionally. According to Sawyer (2003), if a member imposes his or her will on a problem-solving task and enacts a powerful leadership role, it will reduce collective creativity since it limits the other participants' action possibilities and creative freedom. The "space" for expressing new ideas, dissent, or divergent alternatives is thus reduced in this way.

Grant's (1996a) modes of interaction or mechanisms for integrating specialised knowledge, i.e. "rules and directives", "sequencing", "routines" and "group problem-solving and decision making" can be seen not only as means to integrate diverse knowledge, but, I hypothesize, also indirectly as four different ways of constraining diversity.

Rules and directives. Rules, as well as directives, represent impersonal approaches to coordination involving the use of "plans, schedules, forecasts, rules policies and procedures, and standardized information and communication systems" (Van de Ven et al. 1976 p. 323, quoted in Grant 1996a p.114). Grant argues that rules may be viewed as standards which regulate the interactions between individuals. The behaviour of people who follow certain rules is thus guided and pre-determined, which then may result in less room for individual difference in how to carry out a task, or generate new ideas. This category of mechanism also comprises of documents such as schedules and procedures, which are often formal and restricted. These documents have the effect of holding back the participants, or at least constricting personal messages, blending of interpersonal differences, and ways of acting and communicating. This suggestion can be compared to Kellogg's et al. (2006) findings on standardizing media.

Directives involve formalised instructions that are expressed explicitly in policies and manuals for standard operations, procedures or activities; specifications of what, when, and how things should be done. The conversion and codification of tacit knowledge into explicit knowledge in the form of rules and directives "inevitably involves substantial knowledge loss" (Grant 1996b p. 379) since tacit knowledge contains more than we can tell (Polanyi 1966). Such conversions tend to result in documents that are strongly de-personalized and "free" from individual divergence. Moreover, Grant (1996b p. 379) asserts that "[t]he more complex an activity, the greater the number of locations in which that activity must be replicated, and the more stringent the performance specifications for the outcome of that activity, the greater is the reliance on knowledge integration through direction." One example of this is aircraft service and repair work, another can be found in fast food chain restaurants. In these contexts, there is little room for individual variation, inconsistency, and diversity in task accomplishment; unplanned novelty is thus restricted.

Sequencing. As articulated by Grant (1996a p. 115), “sequencing” as a knowledge integrating mechanism involves the organization of “production activities in a time-patterned sequence such that each specialist’s input occurs independently through being assigned a separate time slot.” New product development can entail either sequential or overlapping phases, or fully concurrent phases. The sequential pattern of interaction allows for the integration of specialised knowledge without communicating the knowledge between the individuals. In the sequential way of working, more and more pieces are developed and added, and the product is completed gradually and linearly. This means that project members who receive the product in the late phases of the project (when most of the pieces are set and fixed in place) may have less influence on fundamental design features and construction specification, which, in turn, might imply that their creativity, experiences, and learning are less utilized.

Organizational routines. “An organizational routine provides a mechanism for coordination which is not dependent upon the need for communication of knowledge in the explicit form.” (Grant 1996b p. 379) Routines rely, to a great extent, on informal procedures which are formed through repetition and training. “While routines may be simple sequences, their interesting feature is their ability to support complex patterns of interactions between individuals in the absence of rules, directives, or even significant verbal communication.” (Grant 1996a p. 115). The work of many different teams is based upon routines. “Observation of any work team, whether it is a surgical team in a hospital operating room or a team of mechanics at a grand prix motor race, reveals closely-coordinated work arrangements where each team member applies his or her specialist knowledge, but where the patterns of interaction appear automatic.” (Grant 1996b p. 379) Sophisticated routines permit simultaneity and perfect timing of interactions. In addition, organizational routines can display varied patterns of complex interaction as responses to different stimuli. One can imagine that flexibility, unpredictability, and improvisation in an on-going performance may have significant consequences on the outcome. As a result, organizational routines are also suggested to place a (potential) hampering effect on creativity and diversity.

Group problem-solving and decision making. Even though group problem-solving and unregulated interpersonal communication is supposed to permit and stimulate diversity, it can also be assumed that social interaction and mutual adjustment, which involves and relies on different norms of good manners and politeness etiquette may have a restricting or shaping impact on how people behave. Project members are certainly eager to negotiate, compromise, make joint decisions, and reach consensus sometimes merely to move forward and finish the task, which also may mean that they collectively and personally (and not by rules or directives) constrain divergent behaviour or unconventional ideas.

Finally, we turn to the concepts “interactional frame”, “collaborative emergence”, and “downward causation” as discussed by Sawyer (2003), to suggest how diverse team members’ creativity and possibility to act out their differences is both fostered and hampered by their interaction. The interaction pattern that evolves in, for example, a cross-functional project shapes an “interactional frame” which both enables and constrains creativity and action. The interactional frame is created in a bottom-up process called “collaborative emergence”. Individuals jointly influence and create the interactional frame or context during the development process when they contribute to the task with their diverse ideas and actions. The frame consists of various elements such as role definitions, task and activities specifications, rules, routines and norms, guiding beliefs, and evaluation criteria.

The frame supports participants as they define and make sense of situations, evaluate alternatives and choose how, what, when, and where to add their different ideas, knowledge, and prior experiences. However, while the interactional frame facilitates creativity and action as they are being created, it also constrains possibilities and individual differences. The frame limits actions at an increasing rate over time as more and more relationships form, and the process and product becomes more complex and near completion. It also implicitly determines inappropriate manners and what one should avoid or what does not fit in with the actual situation. Sawyer (2003) referred to this top-down process as "downward causation" since the frame also has casual impact on the individuals' choices and actions. Sawyer (2003 p. 86) states in addition that "[a] more complex frame, once it has emerged, begins to provide some of the power required to maintain coherence across the interaction." The frame thus helps participants to be imaginative and creative, simultaneously as it helps them to keep discipline and on task.

In summary, standardization, time pressure, rules, directives, structure, specifications, norms, and interaction patterns may have a hampering impact and imply less room for the expression of individual difference and creativity. At the same time, as Sawyer (2003) described, strong elements that specify the context and make up an interactional frame also enable coordinated actions. There are certainly many other organizational phenomena, not related to knowledge integration mechanisms, that also might restrain diversity; yet this list will provide a starting point and a general line of thought that will be used in interpreting this thesis' case story.

Generally, the role of knowledge integration mechanisms in various stages or phases in knowledge processes that take place in time-limited development projects has not gained much attention. Hence, it is my ambition to take this literature a step further by investigating how these mechanisms, as well as other collective means, may be used to solve problems and pace diversity and creativity as a project evolves from start to finish. In the following section, the importance of using creativity and knowledge diversity properly, over time, is discussed.

3.3 Pacing diversity and creativity to harness difference

Larson (2007) has examined the importance of knowledge diversity among group members when working with complex tasks. However, he emphasizes that having different types of knowledge is not sufficient; project members must also use their knowledge in a suitable way when it is needed (Larson 2007 p. 414):

> One way that deep diversity can benefit group performance on a complex task is by increasing the range of task-relevant resources the group collectively holds. When different members possess different types of knowledge, skills, and abilities germane to performing the task, the group as a whole has more to work with – and so greater potential to perform well – than when every member possesses essentially the same knowledge, skills, and abilities. However, to capitalize on this potential, it is important that members not only apply their various resources to the task but also that they do so at the appropriate time and in the proper sequence.

As Larson (2007) has also pointed out, when and how project members should use their knowledge is not always easy or even possible to specify in advance, due to the intrinsic complexity and uncertainty of the task. The development context puts strong demands on the team members' collective ability to improvise and coordinate their contributions in an appropriate manner as they carry out the task (Larson 2007 p. 414):

> For many tasks, coordination requirements of this sort cannot easily be specified with precision in advance. Rather, it is necessary instead for the group to coordinate its activity on an impromptu basis through a process of mutual behavioural adjustment as they go about performing the task. Thus, on such tasks, the performance benefits hypothesized to accrue from deep diversity should appear as an emergent property of group interaction.

The challenge to balance, that is, promote and constrain diversity appropriately in relation to the situation or phase that the project is in at a specific point in time may not only be a problem for the project workers to solve, or something that emerges in collaborative processes, but also a matter for the project management to consider and adjust over time.

Söderlund (2010) uses the concept "knowledge entrainment" to discuss pacing and synchronization of different knowledge processes and problem-solving cycles that take place in projects to accomplish knowledge integration and ultimately new knowledge. "We define knowledge entrainment as the adjustment of the pace or cycle of a knowledge process or problem-solving sequence to match or synchronize with that of another to provide processes by which individuals combine their information and uniquely held knowledge." (Söderlund 2010 p. 137-138) This refers to the ability of project members to understand "when, where, and what to do at certain stages and occasions in the project, as well as what to receive and what to send to other participants and units within the project." Söderlund (2010 p. 136). Project members may have different time orientations and work tempo, for instance, "an external consultant working full-time in the project would have a very different tempo compared to the in-house project participants with responsibilities for on-going maintenance", as Söderlund (2010 p. 136) suggests. "Individuals and teams within a project tend to have quite different endogenous rhythms and problem-solving cycles because they are separated in different parts or sub-systems, focus on different domains of the environment, and are involved in problems of different nature." (Söderlund 2010 p. 138). For project managers, knowledge entrainment thus becomes an "everyday matter involving, for instance, the decision of who among a group of conscientious experts is 'right' on a difficult task, what trade-offs or compromises need to be made, and what specification changes are necessary" (Söderlund 2010 p. 137). In such a situation, the project management team may use different time-based control mechanisms, such as milestones and deadlines that create time-pressure and a sense of urgency. This is done to pace different knowledge or problem-solving processes and achieve synchronization so as to "ensure that the 'system-wide knowledge process' – the integration of a diverse set of individually held knowledge – is progressing in an effective and efficient way" as Söderlund (2010 p. 137-138) explains.

Sawyer (2003) compares the stages of problem formulation and problem-solving in a problem-solving process to the "divergent" and "convergent" phases of creative processes. In the first stage, problem formulation or divergence, participants create and formulate a problem and propose many ideas and concepts from their diverse perspectives in a creative brainstorming-like manner. In this phase, the participants do not necessarily relate the proposals, evaluate them, or care for how the different ideas or inputs will work together. In the second stage, problem-solving or convergence, the members evaluate and connect ideas, select alternatives, and jointly work out a solution to the problem, a final product, or concept. Their ideas and contributions thus “grow” and diverge in the beginning of the process, perhaps somewhat inconsistently, to later in the process, where they are restrained in order to converge and cohere. Sawyer (2003 p. 119) states that the members are “solving a puzzle of which they have created the pieces.” Just as a development process may consist of various problem-solving episodes and cycles, it can also display several phases of divergence and convergence. Söderlund (2010) also discusses the timing of knowledge processes and problem-solving activities in terms of creating a project “rhythm” or “beat” to pace contributions, learning, and experiences in a way so that changes can be taken care of and properly integrated in the overall process.

Nonaka (1994 p. 24) also discusses “interaction rhythms” in terms of divergence and convergence, and asserts that the management of an interaction rhythm plays a crucial role in pacing and speeding up knowledge creation processes. “Within the team, rhythms of different speed are first generated and amplified up to a certain point of time and level, and then are given momentum for convergence towards a concept.” Nonaka (1994 p. 24). Whilst creativity is much coveted, it is also necessary to progress in a timely manner, as discussed earlier. Here it is important to know how long a process will last, in order to manage the balance between divergence and convergence, according to Sawyer (2003). Generally, in short projects, the members have to move from divergence to convergence relatively quickly, but in longer projects, the participants can delay the convergence phase and prolong the divergence stage to allow for their differences and use the diversity in the team. This leaves the room open for more creativity; one should “extend the divergence stage so that the trajectory does not reach convergence too early.” (Sawyer 2003 p. 119.). In time-pressured projects, members may be forced to reach convergence early at the expense of creativity.

As one might expect, managing interaction rhythms in practice may not be as straight-forward as it perhaps seems in theory. How this happens and what drivers, mechanisms, or means exist to manage such processes remains to be studied, and also whether “self-organizing” (Nonaka 1994) cross-functional teams invent such measures or mechanisms in collaborative processes or whether specific devices are imposed by the project management in order to appropriately manage phases of divergence and convergence.

4. Summary and research questions

This part concludes and summarizes the chapter. It also reiterates the research questions from Chapter 1 since these may be useful to have in mind when reading the empirical chapter, Chapter 4.

4.1 Chapter summary

This chapter outlined a conceptual point of departure from which the process of knowledge integration and creation which occurs in complex development projects can be studied. I presented an “action-oriented” approach towards knowledge integration and started by discussing *who* integrates knowledge and *how* knowledge is integrated. Two general views on how knowledge can be integrated were presented; through transferring or sharing knowledge, and through the combination of specialized knowledge. I argue that knowledge integration should be conceived as a matter of knowledge combination instead of one of mere knowledge sharing. Following this standpoint, knowledge integration is assumed to be accomplished through the use of different organizational mechanisms. The mechanisms allow different knowledge bases to be combined in novel ways with minimal cross-learning among individuals. Various underlying processes and activities were also investigated. They offered us an understanding of what the knowledge integration process may entail; elaborating on what the knowledge integrators actually do. I also suggested that knowledge integration and knowledge creation should be viewed as two sides of the same coin since these processes are closely intertwined, at least in the context of a collaborative complex development project.

I also discussed *where* knowledge is integrated and suggested that the development process that takes place in a cross-functional project matches the way the knowledge integration process is understood in this thesis. Also in this section, the collaborative and emergent character of project processes was emphasized. The cross-functional development process was, in turn, further explored and conceptualised as a problem-solving endeavour with its different stages of problem definition, solution search, and evaluation of results. This can aid in making the knowledge integration process observable in real-life and, later on in the thesis, facilitate the examination of how different knowledge integration mechanisms may evolve and vary over a development project process. We then turned to the specific issue of unique knowledge and how diversity may be promoted and constrained. Finally, the importance of balancing and pacing diversity and different knowledge processes appropriately during the course of development was discussed.

4.2 The research questions again

To end this chapter I reiterate the research questions stated in the introduction of the thesis. These questions should put the core ideas from the theoretical chapter together and remind us what to look for in the empirical case description that will turn up in chapter 4:

1. What knowledge integration mechanisms and other collective means to enable knowledge integration and project progress can be identified throughout the project process?

2. How are the identified knowledge integration mechanisms and the other knowledge integration enablers used to extend or limit the space used for expressing knowledge differences in different project phases in the face of the approaching deadline?

Before going into the case we will take a look into the making of the thesis; the research approach and method.

CHAPTER III

III. RESEARCH METHOD

Behind the scenes

1. Introduction

This chapter will take you behind the scenes into the making of this thesis. The rest of the thesis is linearly organized and presented which hides a sometimes quite messy process. This chapter reveals how it really was to produce the thesis; the other chapters are constructed and arranged not to mirror the research process but to exhibit the result of it.

At the heart of the study lies one year of fieldwork and several years of material interpretation and generation, accomplished through interplay between my experiences from the field and the reading of literature from different research areas. I will debunk my reasoning throughout the research process, how I have been working, what field strategies I have used and how the empirical material was generated and turned into a research account. This should serve as a guide to understanding the following chapters. The aim is also to give an honest report of how the research and its outcome came about so that the reader can form an opinion and judge the value of the result. The chapter is structured chronologically in order to describe the fieldwork process as it unfolded but first in section two I begin with some background information about my research and ethnography interests, the role that theory played and how the case was selected. The field work description turns up in section number 3 and is divided in three phases; the entering stage, the middle step and the last period before I went back home. Then in section number 4 we will plunge into the desktop work – the processing of empirical material and the writing endeavour.

2. Before entering the field

Before I explain how the fieldwork was carried I give a short introduction in 2.1 to my research interests and ambitions that I had in the beginning of the doctoral studies, just to inform about the background of this study. Then in 2.2 I will present my reasoning regarding ethnographic research. In 2.3 the role that theory played is discussed.

2.1 My research inclination

Long before I knew what the precise research questions would be I had found out that I wanted to do a qualitative empirical study. The appetite for qualitative empirical research was whetted during my time as a student in business administration at Linköping University. Upon embarking the doctoral studies I wanted to "do more" and include observations as well as interviews during a lengthy stay in a company or organization. To do fieldwork and getting a first hand contact with an organization and its people, and create a close everyday understanding of a social phenomenon seemed like a fun and exciting adventure. I would enjoy being out there talking to people, and watch and learn their daily business life, while at the same time continuing the academic work of analyzing, interpreting, imagining, and writing.

Theoretically, I was driven by the desire of exploring something new over which I could ponder and discuss in written and spoken words with the research community, or more specifically, with those interested in research ticketed with organization, knowledge organization, knowledge management, project organization or project management. I did not want to "merely" test theories, hypotheses or what other researchers have presented in different readings. My interest and hope was to offer a new perspective or way of understanding something, or come up with a new notion or concept to describe previously unspecified essentials of human action. To do this I not only needed an interesting case to study carefully but also patient permission to travel around in the literary world to get inspiration and blend various theories in the interpretation of what I would learn from the exploratory odyssey in the field. An open case study would let me generate rich empirical material out of which I then could, through fusion with existing literature, create questions, select focus and perspective and write a "topic-oriented ethnography" (Spradley 1980). This constituted the starting point of the research process. Before presenting the field process as I experienced it we will dig more deeply into the concept "Ethnography". I will also explain the role that theory played in this research as well as how the case was selected.

2.2 Thinking of ethnography

As hastily mentioned in previous section I expected that my fieldwork experience and literature consultation would end up in a micro topic-oriented ethnography (Spradley 1980). The word ethnography comes from Greek; ethnos means "folk" or "people" and graphein means "to write". With its roots in anthropology and sociology, the definition of ethnography can read as follows; "An ethnography is written representation of a culture" (Van Maanen 1988 p. 1) and "Ethnography is the work of describing a culture" (Spradley 1980) and similarly "an ethnography is fundamentally a writing (graphy) of a culture (ethno)" (Harvey 1997 p.212) or as Geertz (1973 p.9-10) with inspiration from Ryle so famously put it "ethnography is thick description". Thick description does not only refer to a cultural narrative rich in detail but more profoundly to a sort of account that contains contextual interpretation of different people's meaning and understanding of situations and actions.

Moreover, Agar (1996 p.53) suggested that ethnography can be "both a product and a process" where the product is the written account of the life of a particular social group and the process is the fieldwork undertaken to learn about that particular group. Prasad (1997 p.103) assumed that ethnography is mostly understood as a *qualitative method* with a set of principles and techniques but asserted that it also constitutes a *perspective* on different phenomena such as technological change, implementation of a new information system, or organizational change. The ethnographic perspective would then imply that social phenomena are interpreted through cultural aspects such as symbolic action, rituals, ceremonies, myths, and heroes. Moreover, ethnography constitutes an exploratory *research approach* for research undertakings that aim at creating new understanding of unknown social situations from the "natives'" point of view (Prasad 1997). Furthermore, Prasad (1997) argued that ethnography could be seen as a *methodology* with certain ontological and epistemological beliefs rooted in symbolic anthropology and in Geertz' (1973) "Interpretive theory of Culture" in which the concept "thick description" is pivotal. Actually, according to Geertz (1973 p.10) things' ontological status is irrelevant, what counts is their meaning; "The thing to ask is what their import is: what it is, ridicule or challenge, irony or anger, snobbery or pride, that, in their occurrence and through their agency, is getting said." He argued that the core of culture analysis is interpretation of meaning since founding culture analysis on mere observation of social actions, as advocates of behaviourism propose, will not suffice to create meaningful understanding of situations. The epistemological assumption concerns the primacy of local knowledge over universal laws and grand theory as paths to understand the world, "small facts speak to large issues" (Geertz 1973 p.23).

Even though I found ethnographic research inspiring and adequate for investigating what I was concerned about I did not seek to describe the total way of life of a people or an entire culture. I was not attracted by or interested in using the cultural terminology to analyze and interpret what I experienced in the field. However, I could still use the content of the words *"qualitative ethnographic method, ethnographic methodology and ethnographic research approach"* explained above (Prasad 1997) to describe my research. I have used ethnographic techniques and principles (*method*) which will be explained under subheading number three in this chapter. Furthermore, I adhere to the notion that social actions must be interpreted in its context, or as Geertz (1973) put it, in its "webs of significance" and that what counts is the meaning that different actions have for different people. Moreover, to be able to say something little about something greater one needs to start in the "local village" (Geertz 1973) and then take it to the "bigger world" (*methodology*). There in the "local village" I strived for learning about the business from the diverse natives' different point of view to be able to generate "thick description" (Geertz 1973). Van Maanen (1988) argued yet that it is one of the limits with ethnography; it produces local knowledge and only marginal contribution to an area that is focused on and interested in grander issues. It might become hard to identify patterns in such a small setting and if one does, the pattern runs the risk of being quite unique for that particular context. Nevertheless, Spradley (1980 p. 16) assumes that "ethnography offers one of the best ways to understand….().. how people with diverse perspectives interact", which is the focal area of my interest and Moeran (2007) among others also asserts that ethnography suits when researching diverse perspectives and also when the research has exploratory purposes. So, with an exploratory attitude (*research approach*) I expected to see things or aspects of the topic that had not been discussed in the same way before.

Historically and traditionally, ethnography was what cultural anthropologists did when studying social tribes and "exotic" cultures in alien (to the researcher) worlds by observing or immersing into the tribe's way of living. The total way of life, or selected cultural aspects of it, such as rituals, ceremonies, language, symbols, artefacts, and social behaviours were examined, and reported back to the homeland. Today, according to Van Maanen (1988), Spradley (1980), Prasad (1997), and Garsten (2004) among many others, modern ethnography is conducted by diverse researchers across various disciplines. Different social processes, activities, phenomenon and groups in various settings and situations, both near the researcher's neighbourhood and in foreign places, are studied ethnographically. Basically, any human arena constitutes a target for ethnographic research (Van Maanen 1988, Spradley 1980, Prasad 1997, and Garsten 2004).

Spradley (1980 p. 30-31) named a kind of ethnography concentrated on a selected problem or topic in a specific "single social situation" as "topic-oriented" "micro-ethnography", which seems similar to what I have done. In my case the topic concerned knowledge integration, diversity and problem solving. The "micro-ethnography" which focuses on one "single social situation" can be defined by three elements, Spradley (1980) suggested, namely, "place, actors, activities". In this study the elements were a cross-functional development project at PPM in Stockholm and Söderhamn, Sweden (place), project team members and project managers (actors) and the daily project and development activities (activities) that they carried out.

The details of the research agenda changed underway in accordance with what I discovered and learned. As Spradley (1980 p. 26) stated "The ethnographer has much in common with the explorer trying to map a wilderness area" in the sense that the researcher in both cases adjusts and specifies the research focus during the way instead of following a predefined linear research plan. According to Spradley (1980), Agar (1996) and Alvesson and Kärreman (2011), the researcher should not entry the field with precise questions (Spradley 1980 p.33); "Instead of coming into the field with specific questions, the ethnographer analyzes the field data compiled from participant observation to discover questions." The researcher should let him/herself be surprised by the empirical material as Alvesson and Kärreman (2011) discussed. The purpose and questions were thus only vaguely defined at the outset to discover whatever the field would carry. I did not want to lock myself into a specific view or only seek narrowly after something defined in advance and jeopardize that crucial elements of the social situation that could constitute the new stuff to theorize about were missed out. I used some sort of funnel approach; a broad start and an open mind with questions oriented towards understanding; who were the actors, what was going on, why, how, and where. I wanted to learn about the people and their differences, PPM, the pension system, IT, projects, people's, relationships, activities, language, problems, routines, conflicts, their way of talking and doing things, their tools, beliefs - their way of living together in the organization and so get a feeling for the situation. Then later on in the process the study was delimited and particular themes, perspective and aspects were selected.

Nevertheless, I had of course an idea what to study (and why) already at the outset, which was based on my previous experiences and background knowledge and the tradition and research focus in the research group. Described in Karl Popper's terminology, the process was guided by a "searchlight" consisting of a tentative purpose and my personal knowledge and background. As various scholars argue the researcher's personal background and social context influence deliberately and unconsciously what is considered important and relevant to study and focus on (Agar 1996, Alvesson and Kärreman 2011). The way the selected aspects are interpreted depends also on the pre-understandings of the researcher and what is considered interesting and good research in the research village at home (Agar 1996; Alvesson and Kärreman 2011). Fieldwork, according to Geertz (1973), Agar (1996), Van Maanen (1988), Alvesson and Kärreman (2011) and Garsten (2004) is a process where data is generated through interpretation rather than objectively observed and transparently and neutrally reported – behind is always interpretation. Empirical work is thus very much a construction process (Alvesson and Kärreman 2011) and an interpretative act (Geertz 1973). We will come back to how I dealt with my background and potential bias further down the chapter.

The research area and questions have not only been adjusted to the learning gained from the field but also through interplay with theories and various readings. The open funnel approach concerned not only the empirical work but also the theoretical study. The interplay and the role that theory played is discussed in section 2.3.

2.3 Role of Theory

Beside the aspiration to create "thick description", that is, comprehensive understanding of a social situation incorporating different people's perspectives, I had also a theoretical ambition, as stated in the beginning of this chapter, to develop theory with a new understanding or new concept. Theories and empirical material were used alternately in an abductive way and allowed the understanding and the new concept to grow gradually. I started in theory, which influenced what was interesting to pay attention to out in the field, and then the experiences and learning from the field encouraged new theory search and further reading, which in turn made me interpret the field experiences in new way and also "detect" and consider new things in the field. And so this process continued until the last word of the thesis was written. Inspiration from theories in combination with empirical material can stimulate theory development and the creation of new concepts according to Alvesson and Kärreman (2011). They argue that a fruitful way to theory development includes problematization and critical discussion towards established theory. The researcher should not only seek for "gaps" in existing literature but also question underlying assumptions and understandings. In their perspective, empirical material can be seen as a "critical dialogue partner" (Alvesson and Kärreman 2011) that could help challenge and criticize existing theory and problematize assumptions and so "speak back" to our pre-understandings. It constitutes a source of inspiration to invent new concepts or understand current concepts differently and so produce more imaginative studies.

Although a critical approach with ambition to reject some existing assumptions and put forward a completely different understanding and revolutionary theory may be attractive and perhaps a researcher's dream it would not only be over-optimistic for a doctoral student (and not recommended as Alvesson and Kärreman 2011 said) but also reveal a certain amount of hubris and naivety. What I believed was possible and little more down-to-earth was to go through the literature to see what was missing in current writings, that is, to start with identifying a gap and then further down the process end up with a more challenging discussion against existing theory. In the end I would confront with a new understanding, concept or theory in miniature that could mobilize a minor part of existing theory. The role of theory here was thus to find something to build on and avoid reinventing the wheel, explicate the theory further and see it from a new angle. Even though I wanted to keep the field process open to see and learn whatever there was I could not let myself go completely wild since I would risk getting lost so theory constituted a way to discipline the fieldwork, which is a well-known use of theory (Alvesson and Kärreman 2011). In a process of "disciplined imagination" in Weick's (1989) words, Alvesson and Kärreman (2011) recommend researchers to emphasize "imagination" if the ambition is to create interesting research.

In particular, I argued that there should be some room for more ethnographic process studies of knowledge integration to understand how knowledge integration mechanisms might be used in portfolio and perhaps change over time. As the gap was being filled (I identified different knowledge integration mechanisms during the different stages of the project) I realized that the use of knowledge integration mechanisms could not alone describe what was going on. Early in the research process I also tested the concept "boundary objects" but it did not either seem perfectly appropriate here. Moreover, project management and project organization ideas helped me create understanding at an overall project level but could not either assist in pinpointing the essentials of the process and its critical situations. Furthermore, discourse and communication theories were tried out since the information systems development to a great extent was intangible work with only little physical tools and equipment in use. Much work was carried out in interaction and communication; a great deal of a workday for many project members consisted of meetings, so it seemed logical to analyze their verbal interaction, the everyday talk. However, discourse analysis and communication theories could only facilitate my interpretation to some extent. One problem was that the interaction and the events that I observed or participated in were problem solving-oriented and goal-oriented, an important characteristic that many of the communication studies that I consulted did not focus on which made them hard for me to use. Also, studying language per se in this complex setting was difficult since it was deeply context dependent. The meaning of the spoken words could not be understood unless the context in which the words were said also was studied and interpreted. This leads to a reflection on the role of contextual understanding.

The context was not only important to consider for me personally. It was also important to explain the context to my audience (research fellows) so that they knew why and how things had been interpreted or how conclusions were accomplished. The figure below comes from Kahneman (2011 p.79). It constitutes an example on how the context influences interpretation. (The original figure has been amended. It first contained two examples showing the same point. I have removed one of the examples in the middle of the figure and saved one.)

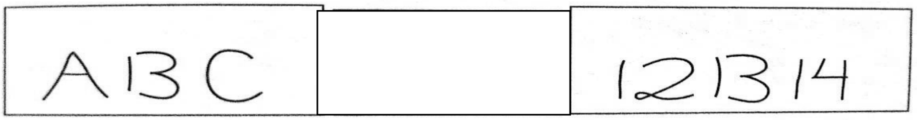


Figure 2: The context influences interpretation

The reader presumably understands and reads the displays as “ABC” and “12 13 14”. However, the central shapes in both boxes are identical. Nevertheless, the reader probably interprets the first as the letter “B” and the other as the number “13” even though one could as well have read them as “A 13 C” and “12 B 14”. But “The same shape is read as a letter in a context of letters and as a number in a context of numbers. The entire context helps determine the interpretation of each element.” as Kahneman (2011 p.80) explained. On the one hand the context helps to make sense out of things and reduce ambiguity in different situations. On the other hand it may lead us to draw conclusions too quickly, disregarding alternative interpretations. The best one could do seems hence to try to understand the context and explain particular events in the context and produce alternative interpretations in relation to how the context is interpreted.

However, I found that the vocabulary of knowledge management and organization, boundary objects, project management, and communication studies could not account for what occurred in the field. I needed some other tools to make sense of the situation. I continued to read and came across some interesting work regarding small group research and diversity, problem solving and behavioural economics. Through fusing some ideas from these theories with other more familiar research I generated an interpretation of the case project that in turn could be used to develop a new concept. The new understanding and the new concept could then be applied in a more challenging discussion towards existing theories of knowledge integration. These tours through different theories were intertwined with different interpretation turns of the field material. The process is similar to what Alvesson and Kärreman (2011) referred to as working with “breakdowns”. A breakdown occurs when the researcher encounters a situation where his or her expectations do not harmonize with what is going on in the studied culture. The researcher becomes puzzled and has trouble understanding and explaining the situation. The researcher dissects the ingredients of the situation and changes the research plan. Breakdowns continue to happen until the researcher understands the culture satisfactorily. Working with breakdowns becomes in this way a part of the research method.

A final note in this section concerns the theoretical chapter of the thesis. It serves the purpose of showing the reader what I consider important and relevant to know before reading the case. It is supposed to sensitize the reader to look for certain things in the empirical chapters.

Furthermore, the theoretical chapter constitutes a starting point rather than a theoretical framework or complete literature review. It denotes to which research field the contribution is given. Now, before describing the fieldwork as it unfolded, I explain how the research site and case project was selected.

2.4 Selecting a case

When I started searching for a good place to do fieldwork I discussed my research interest with friends and colleagues whenever I could and wherever I went. The response from people in general was that they recognized the problem (knowledge integration in terms of collaboration difficulties among diverse people in development settings and different means to bridge such knowledge-related differences and solve collaboration and communication problems) and became interested in the research. Many of them got enthusiastic and said that I should come to their office to do the fieldwork. However, most of the organizations were not suitable. One of the people I met though, an alumni from Linköping University (industrial economics), worked at the Premium Pension Agency, PPM. This organization, especially one of their projects, a business development/information systems development (ISD) project, turned out to fulfil the selection criteria, which included people and their knowledge-related differences, project organization and activities, size, time, place complexity, accessibility.

First, the project members of the chosen case project had different educational background, expertise and work experiences, which already had collided and caused serious collaboration and communication problems that they needed to solve in some way in order to succeed. The diversity among the organizational members was the first and perhaps most important criteria to be fulfilled since the study's overarching theme was knowledge integration. To be interesting as a case people should have different backgrounds in terms of education and training (for instance IT, Insurance and Law) and belong to different functional units (IT, communication, insurance, law), be trained in different "schools" or work methods (e.g. project management, software development methods), and traditions and industries (e.g. government agencies or management consultancy). The project members should not have spent much time working together previously either, at least not all of them, since working closely together may make people similar to each other and turn them into a "tightly knit group" with strong communal knowledge base.

Moreover, the project had a goal and estimated time (and budget) frames with a start and an end which made it easy to frame the empirical setting. Also, their time plan suited my research situation. It was important that I could stay at their place during this period. Agar (1996 p.120) argued that "First of all, it takes a while for people to accept your role and begin to trust you. Then to achieve the kind of learning to which ethnographers aspire, much time is necessary. People have different sides of themselves that they display under different sets of circumstances, making it essential to see group members in different situations, not just during brief interview." Moreover, Agar (1996 p.120) discussed "For the same reasons, there is also an emphasis on the ethnographer going into the group's home turf to do the research. People are usually more comfortable in their home territory, compared to bringing them into an office or laboratory, though there are times when an ethnographer needs a quiet place for personal interviews. Then, if one is interested in all the situations that a person ordinarily moves through and deals with, it only makes sense to be there when it happens. Finally, because much ethnography can be translated as becoming part of a group, living with them is a usual correlate of being a part-member."

Furthermore, the project had a considerable organization consisting of subprojects, project managers, a project board committee and a project office but was not too big or complex to use as a case project. Around 30-35 people worked in the project. Project activities were mostly located in Stockholm but also in Söderhamn, app. 200 kilometres north of Stockholm. The location was appropriate for me since I could afford staying there for a long time. I could also easily travel back and forth to Linköping as much as needed (app. 200 kilometres between Stockholm and Linköping).

Moreover, the accessibility was of course an important aspect in selecting a case. At PPM and in this project I had the opportunity to be in the middle of the scene at the "central court" and follow the process from "inside". I could come as often as I wanted (and go whenever I needed). I was entrusted with a pass card, personal desk, and email address and was allowed to observe meetings, keep records, dig in the electronic archives and read documents, surf around on the intranet and interview and talk informally with different people at all different departments and hierarchical levels. However, there were some access restrictions in the beginning due to the time pressure that was put on the project members, especially the developers. I was not allowed to sink their work pace with interviews or questions or anything that could disturb their work. This restriction was soon lessened and I got many chances to talk with them in between meetings and in interviews. Later on in the process I participated in certain activities but I mostly observed, talked and interviewed people. Possibilities to participate may be important selection criteria, according to Spradley (1980) but as Agar (1996) said, the ethnographer is probably not a great help in a complex setting anyway since the activities often require more specific knowledge, skills and task training than the researcher has. I could not work on the systems design or write program code but I participated in some testing activities. More importantly, I assisted in evaluation of the project after project completion as well as in reflection seminars and presentations during the project process. The evaluation focused on project organization, collaboration and communication aspects. All this will be discussed in the coming parts.

Moreover, one could also expect that the knowledge integration "problems" should turn up several times in different phases throughout the project, that is, the phenomenon to be studied was expected to recur in different ways and through different activities, which Spradley (1980) also said could constitute an important selection criteria. The task was unique and non-repetitive and most of the activities were of development character with much innovative thinking involved and constituted a first time experience for the project members. The task was complex also in the sense that it involved interdependencies among parts and interconnected systems. Moreover, project members did not only have to find out how to solve the task technically but also organizationally since they had not worked together in the same constellations before. No one knew how the project would end, what the final outcome would look like, what problems would turn up during the way or if they would make it at all. In addition, the outcome of the project was extremely important for the organization as well as indirectly for all Swedish pension savers, which placed the project in a prioritized position at PPM and people put a lot of effort into solving the project task in a good way. Thus, along with a strong time pressure this seemed to be an interesting case to follow.

The fieldwork started with a meeting at PPM where the Head Project Manager and the financial controller of the project introduced the project and presented its background and where they were at the moment in the project process. I presented my research proposal and my advisor spoke about our research group and some previous research projects. Fortunately, I was directly welcomed and could start the fieldwork right away.

3. Dwelling in the field

I entered the field in November 2003 and stayed until December 2004. This part of the chapter, number 3, has three main subheadings. The first one, 3.1 “Entering”, discusses the first period in the field, November 2003-February 2004, and the different field techniques or strategies that I employed during the first months. The next section, 3.2 “In the middle of the process”, describes how my involvement with the project intensified and the dilemma of being an insider and outsider at the same time. The middle of the process lasted from springtime 2004 until September 2004. There were thus some breaks during the process (teaching, conference and project’s summer vacation) but during these periods I kept myself updated on what happened in the project. Before going back home I participated in the project’s evaluation procedure, which occurred in October-December in 2004, plus two visits in January and February 2005. The last phase is explained in section 3.3 “Before walking out”.

3.1 Entering

“…the more you know about a situation as an ordinary participant, the more difficult it is to study it as an ethnographer. (…) The less familiar you are with a social situation, the *more* you are able to see the tacit cultural rules at work” (Spradley 1980 p.61-62)

This cheered me up a little bit since I was a little anxious when I first entered the field well aware of the fact that I knew nothing about information systems development. Nevertheless, I had some background in project organization and project management which seemed to be somewhat similar to what was going on out there in “real life”. However, I had been in different unfamiliar situations before. The feeling reminded of the one I have had on the first day on a new job.

Section 3.1 describes the first phase of my fieldwork. It contains the welcoming and forming of my role in the field and the “field strategies” that I developed in order to understand and learn about the project, its members and PPM.

3.1.1 “Welcome! But please, don’t disturb!”

I came to PPM in November in 2003 and was very well received and welcomed. I was given a personal pass card and my contact person showed me an office at the top floor which I could use as mine. There was a computer with passwords for me and a telephone. I got an email address and one person from the HRM department came to make a photograph of me to put in the digital telephone register on the intranet.

I was introduced to several project members from one of the project groups (The requirement subproject who mostly consisted of data operators/expert end users) who in spontaneous informal conversations nicely and gladly told about the project work so far. Furthermore, I was greeted by the Head Project Manager two stairs down at the IT department. He first gave a background presentation and quick update before he introduced me at an IT-meeting the same day. He announced to the project members that they had a “noble” visitor here, a doctoral student in business administration and project management from Linköping University who would hang around for a while and study communication and collaboration and project management. “But” as he hastened to say “she will not disturb, you will hardly notice her, she will just join the meetings and sit here as a grey eminence”.

After this quick and somewhat facetious but nevertheless serious presentation I did not dare to say a word. I had been notified that the developers and designers were under hard pressure. The development undertakings had not succeeded so well recently and now both time and quality pressure was high. In addition, the PPM process had been covered and criticized in media and many people at PPM were fed up with intimidating journalists. So it was important that they knew I was not there as a journalist to scrutinize or criticize their work and degrade them in public, or take up any of their precious time. As a consequence, I agreed by nodding and promised not to disturb.

I started to figure out how I could reduce my visibility and mingle unnoticed, so that the project members did not spend too much time wondering about my presence and reasons for being there. While unobtrusiveness (Spradley 1980) and a low profile may bring some advantages I still needed to learn and talk with the project members. I could not either conceal my presence completely since I was within a delimited organization with specific organizational members, in a specific physical building.

As regarded my attendance at meetings I thought I better present myself shortly to new (for me) people right before a meeting started and then join them at the table instead of sitting behind starring at their neck, so that they did not have to spend energy thinking about who I was. Nonetheless, it was not always possible to introduce myself before the meeting. It depended on when they arrived, that is, if there was enough time to say something. I never interrupted a meeting once it had started to introduce myself to someone entering a meeting in the middle of it. Frequently, curious project members stayed after the meeting to ask me questions and I also then took the chance to explain who I was and what I did and asked and talked with them to get to know what I wondered at the moment. Nevertheless, I did not interrupt developers in the beginning; I just made observations, unless they spoke to me first and showed me that they had time and interest in talking, which in fact happened several times a week. Even though I was not permitted yet to take up their time in interviews I learned a lot by listening to their conversations in meetings. And as Agar (1996 p.120) argued “You can’t specify the questions you’re going to ask when you move into a community; you don’t know how to ask questions yet.” That was particularly true for me so I was glad that I had the opportunity to sit quietly and just listening.

PPM had a small canteen (more like a kitchen) where one could bring food and enjoy it with others, have a coffee and chat with colleagues. In the beginning, I appreciated to have lunch or coffee there only if I had specific company or a “lunch date”. Otherwise I felt like I was eavesdropping and disturbed people who needed a break from their work. If I was alone, waiting for a meeting or waiting for something special to happen that I could observe, I felt more comfortable to hang out by my desk or in a nearby meeting square. I usually worked on a paper, wrote or read something or organized my notes quietly to avoid drawing too much attention. In this way people could approach me, which they often did, instead of me troubling them. However, as time went by and I made more and more friends I felt also more comfortable hanging out anywhere.

Moreover, I was also introduced to the project office one of the first days. Here, I was allowed to talk as much as I needed with the personnel and read whatever project documents I wanted. In this early phase of my fieldwork process I tried to understand the situation from an overall or broad perspective, to get an overview of the situation; the history of the project, what had happened so far, critical events, big changes, turning points, groups of people and their project managers or group leader, main activities and their relationship and simultaneously try to understand what was going on right now. However, these general descriptions were almost at once also combined with tentative analysis and interpretation. Spradley (1980 p. 76) argued that descriptive observations are suitable in the beginning of the process when the researcher's level of knowledge of the culture is low. I was in this phase on "Grand Tour" in Spradley's words (1980 p.77) which implied that I was searching without detailed questions or specific focus for the major features of the social setting, in terms of "place, actors, activities, objects, acts, events, time, goal, feeling" which resulted in comprehensive descriptions and a growing understanding of the project and its people.

In addition, one of the subproject managers happened to be very interested in my work and enjoyed discussing different things every day I was there. She was also happy to tell me about her experiences from her time at PPM. I was sometimes also asked to go out for lunch or pick up something to eat in and important things (as well as personal chat) were naturally discussed at these occasions as well. In addition, people advised me to contact and talk to different people and asked me to join in at different meetings all the time, which according to Agar (1996) is a sign of acceptance of the ethnographer. Sometimes I almost needed to run between meetings or choose which meetings to attend when meeting time collided. In the beginning I needed the courage to ask "strangers" if I could join them in different meetings and settings; I could not only await invitations since I was worried that I missed crucial events and discussions.

The Head Project Manager continued to keep me updated (on his own initiatives) and let me join him around to participate as an observer in different meetings and problem discussions. Furthermore, as a positive surprise he suggested that I could carry out individual interviews with three subproject managers and two key persons already during the first weeks. He helped me making appointments and reserved a small meeting room where the interviewees and I could talk in privacy. Each interview lasted for approximately one hour. The two key persons who had worked as project managers during the project's first year discussed the first year's troubles and issues. Since I had not been there in the field during the project's first period this became a retrospective part of my study. I continued the retrospective study informally with different people throughout the whole fieldwork process. As I got an understanding of the project's history and its different people I understood more and more of current issues and could so ask more intelligible questions in between the meetings. People gradually looked more relaxed and comfortable with me. Or perhaps it was the other way around – maybe it was me who looked a little more relaxed and appeared little more comfortable! However, I engaged more and more in corridor conversations and talked to different people to and from meetings and by the coffee machine and people often wanted to show me different documents and regularly I got emails with different kinds of information.

Taken together, all this helped me to quickly naturalise and feel comfortable in my role as a "professional stranger" (Agar 1996) despite "the grey eminence status". As a matter of fact, I never felt like a grey eminence (whatever it feels like). Mostly, people seemed to think it was okay and even quite interesting and fun that I was there, which of course was a mutual experience. This is however not unusual – fieldworkers are often more viewed by the community as a peculiar but perhaps entertaining supporting character or "hopeless dummy" (Van Maanen 1988 p.2) than a regular full-time member or participant (Agar 1997; Alvesson and Kärreman 2011). After a month or so I started to feel that I was “someone” and got a sense of belonging even though I did not yet participate in their activities.

3.1.2 “Fake it until you make it” and other early field strategies

During the fieldwork process I practiced some strategies to learn about the project and the project members and their problems. Some of the techniques were thought out on beforehand, at least to some extent, and some emerged in situation as a response to what was needed. These *early* field strategies included “Fake it until I make it”, “Talk about my research interest to unlock the situation”, “Write down the questions and they will be answered”, “Listen to everyday talk and expressions rather than technical details” and “Use the recorder as a memory-extension complement rather than as a substitution to the notes” and finally “Use a mix of fieldwork techniques”.

The first strategy “Fake it until I make it” emerged as a response to the “grey eminence” status given to me. I did not simply want to disturb too much. I let people talk without too many “stupid” questions that would take ages to answer. In this way I hoped to get accepted quickly. I pretended for myself and others that I understood and tried to make sense of things and made up “coherent” stories early. These were stories that I kept for myself. I had of course to revise them many times until I really could describe and explain what I had pretended to understand. And in fact, I believe one may learn in this way if there is enough time and patience to tolerate uncertainty. Tolerate uncertainty means as Agar (1996) has discussed that the ethnographer often does not understand what is going on around him or her but needs to bear that frustration. So what I did was that I looked happy, listened carefully and interestingly to people. I imagine that this strategy works when the ethnographer’s degree of participation is limited to what Garsten (2004) calls “observer as participant”, which means that the researcher does not really participate in the activities that the “natives” carry out but rather observes the people and their activities from near distance and hangs out with them or *lives among* (Rennstam 2007) them rather than lives as them. Then, as when learning a new language, I suddenly realized that I in fact understood the terminology, the discussions and the things they talked about without really knowing when or how it happened.

The second strategy “Talk about my research interest to unlock the situation” concerned the interview situation as well as different informal talks with different people in the beginning of the research process. Here I found that people in general became more relaxed if I talked about my interests first instead of incessantly pumping questions on them. I let them get to know me in order to gain trust and reduce potential asymmetry in the relationship (Agar 1996); that is, I let them understand that I was only a student with some interests that would not harm them personally or as a group and as a matter of fact, I was actually there to learn from them. When people understood what I was interested in they could often relate to themselves or a familiar situation. In this way they picked up the issue and from there we could generate something in between good semi-structured informative discussion and informal conversation or informal ethnographic interview (Agar 1996). The members’ commitment was spurred through stimulating curiosity and interest in the research area, which is also mentioned in Alvesson and Kärreman (2011) as part of the “interview as a mini-seminar” strategy. I practiced active and positive affirmative listening, where I “paraphrased” (Agar 1996) repeatedly to make sure that I understood their point of view. I tried to “dejargonize” (Agar 1996) even if I talked about my interests, and ask broad how-questions to show my deepest interest in their business and use a non-academic language, not only to “make friends” but also as a way to keep the process open since there might be other more interesting problems to look at than those that I could imagine; what if knowledge integration does not occur, what if knowledge integration mechanisms are not involved. I did not want to risk that “informants” adjusted their experiences or made up information to make them fit in the research agenda or to “please” me.

The third strategy “Write down the questions and they will be answered” is associated with the first one. There was so much information and things in the situation that I did not understand in the beginning. Almost everything was tagged with a question mark in my head. Questions popped up all the time. If there was no possibility to ask right away I put down the questions on paper to remember to ask when I got a chance. Actually, I was writing a lot all the time; meeting notes, things I had learned, interesting things people said (often word by word), what I have seen, heard, been, things I did not understand, questions, interpretations, thoughts of different kinds – everything that crossed my mind. However, I noticed that as time went by many of the questions had become answered in some way or another as the process proceeded. I did not have to ask about everything, I still learned somehow from being there, attending meetings, listening to people, and having spontaneous conversations. Perhaps, I stimulated the brains to listen for the answers without really paying attention by writing down questions and thoughts on paper. However, things that remained unanswered and still appeared important could easily be recalled and picked up from the notes to be asked when appropriate occasions turned up.

The fourth strategy, “Listen to everyday talk and expressions rather than technical details” constituted a way to sort in the deluge of information and facilitate understanding in the beginning of the fieldwork. I commenced the research process by trying to grasp current problems and important issues and mapping who worked where and with what and with whom. Understanding started to grow through the way people talked about other people, problems, and events. Like a child I listened to the strength in people’s way of talking and so understood if something crucial was discussed that could be interesting to follow-up. Very low intensity in the interaction and communication with short or no answers could also point at critical problems and issues.

Moreover, metaphors, analogies, and everyday expressions got my attention and explicit awareness and pointed at the relative importance of different things. For example, someone explained “It’s a problem, we cannot change it because it is in their bones” I understood that it was a question of an unwanted habit and could start my “exploration” of the problem from there. Or the comment “I have tried to ask but I can’t reach them” pointed at a communication problem that sounded interesting to dig into, or a manager’s description of a specific team “They are like a bunch of school boys” made me curious about how that group worked, or when a project manager said about the team members “If they say something that‘s the way it will be, I can’t say against them” I wondered about their interaction and the role of knowledge in that situation. A “hot potato” seemed also interesting to understand and constituted a trigger point for further investigation, or “that is our common problem, we are in the same boat now” made me wonder who is in that “boat”, were there two boats before and so on. “They are like artists” and “do we become your hostage now?” and such comments evoked further questions. However, sometimes it has shown to be a non-issue that I sooner or later could just drop. As I understood more of the technical details of the project I could listen more sophistically to technical expressions and analogies as well.

The fifth strategy, “use the recorder as a memory-extension complement rather than as a substitution to the diary” means that even though I recorded as much as I could I still took notes and wrote down things during the meetings. I was very happy about the recorder since I knew that I did not understand and let alone remembered everything in the beginning and would need to go back and listen to the meetings later on. Then, several months later when I listened to the recorded meetings from the early fieldwork I could suddenly make sense of the discussions. Also, my notes were anything but "neutral" so the recorded material constituted my "purest" data that I could return to if I needed to start all over again and create another new interpretation or alternative perspective. The recorder just recorded the information as it appeared while I constantly "processed" the information and worked actively on the understanding and interpretation of it. My notes contained events, date, time, people, places and conversations, things that project members said; sometimes I wrote down word by word to be used as quotes. My notes also entailed my experiences, questions, thoughts, confusions, problems, ideas, tentative interpretations and reflections. I did not use any specific principles for my notes; I just wrote things down as they came to my mind.

It happened every once in a while that it felt uncomfortable and unethical to keep the recorder on, for example, if I realized that I had not introduced me and my research objectives or had the time to ask for permission before a meeting with "new" people. In these situations I turned it off. I did not use it in informal conversations or settings either, for example during lunch time or by the coffee machine. However, if I had the permission at the beginning of the meeting and someone arrived late I did not turn it off but decided that if I wished to use that persons' utterance I would of course check with him or her first. All in all, I was relieved that I could use the recorder so much since I knew I would use many quotes in the empirical text and that could barely my personal memory keep and supply with. I first felt little uneasy with the recorder but as time passed and the project members and I got to know each other, the feeling vanished. The recorder and my presence did not seem to bother people except for one time during a heated discussion. One project member seemed uneasy or uninterested in participating in the study but did not ask me directly to switch the recorder off but in this case I turned it off anyway. As Spradley (1980 p.22) said "No matter how unobtrusive, ethnographic research always pries into the lives of informants. Participant observation represents a powerful tool for invading other people's way of life. It reveals information that can be used to *affirm* their rights, interests, and sensitivities or to *violate* them. All informants must have the protection of saying things "off the record" that never find their way into the ethnographer's field notes." However, mostly people did not seem to care much about me or the recorder. A funny thing here is that the project office controller got inspired and actually bought himself a recorder to use during meetings in order to be able to write better meeting notes!

My "ethnographic record" (Spradley 1980) consisted thus mostly of field notes and the tape-recordings. I brought my recorder, a notebook, and a pen to all meetings and wrote down my observations and thoughts during the meetings and continued afterwards or at home or whenever something turned up in my mind. Agar (1996) is negative to field notes for several reasons. First of all, in the beginning the researcher does not know what is relevant to write down. In addition, if the researcher does not write it down immediately he or she will not remember it. In addition, things happen much faster than we can write so one must rely on a long-term memory which does not work as good as we think, he argued. We tend to remember stereotypical conceptualizations of events rather than specific details. "In their worst form, they are an attempt to vacuum up everything possible, either interrupting your observation to do so or distorting the results when retrieving them from long-term memory." (Agar 1996 p. 162) However, Agar (1996) recommends writing a personal diary which emphasizes the ethnographers' reaction to different things that occur in the field, personal feelings, appreciation of how things proceed and similar. This kind of personal account brings the "ethnographer's role more explicitly into the research process" and could constitute an important part of the ethnographic method, Agar stated (1996 p.163). I combined field notes and personal diary into the same account since that was the way to create understanding and learn about the project for me. I like to write because I actively learn when I write. And since much time was spent in meetings sitting around a table in an office I had always opportunity to write. In the end, which I will return to later on in this chapter, I became so involved in the project and learned it as it was a part of me and my knowledge and experience so that I never really felt that I drown in notebooks or information that I had to read several times to "remember" and I believe that much of this "learning" was related to my diary/field note writing.

The sixth strategy, "Use a mix of fieldwork techniques", is an overall strategy that entails elements from all the other strategies. The activities of this strategy were inscribed in the research plan from the beginning and were expected to complement and strengthened each other. I combined concurrently interviews, observations, participation, document reading, informal talk, diary writing and just "hanging out". I mixed the techniques for various reasons. The overall purpose was that I as quickly as possible wanted to catch up and understand what was going on to in the end get an insider understanding. I used different techniques in concert to see things from different perspectives, stimulate all my senses and in this way boost experience. Using different techniques at the same time echoes stronger and results in a more intensive experience than if I should have concentrated on one at the time.

I was particularly interested in studying the project members' interaction and communication and reasoned that I should observe their real-time real-life ongoing interaction and communication rather than merely interviewing people about it. Project members discussed and even carried out much of the project work in meetings, workshops and seminars, which pointed at the importance of observing those activities. I entered in the middle of the project process and attended all meetings I possibly could. In the beginning I mostly shadowed the Head Project Manager and followed him around but as I soon got to know more project members I also joined them to see different places and observe and participate in various meetings, workshops, hallway or desk-talk, lunch, and coffee breaks. I wanted to see and hear with my own eyes and ears and get a feeling for the situation based on my own experiences from "being there" in Geertz' words (1988), which often is said to be the most prominent feature of ethnographic fieldwork (Moeran 2007; Van Maanen 1988; Agar 1996; Garsten 2004). Ideal characteristics of fieldwork according to Moeran (2007 p.117) include intensive participant observation, "being there" and social immersion (Geertz 1988) which means that the researcher lives with and as the social group that he or she studies. However, when the fieldwork is carried out in an organization, which is inhabited by people only for some hours a day the researcher does only "live with them and as them" during these hours (a study of a certain group of people living together 24/7 would imply another sort of immersion). I did not hang out with the project members after work hours and did not meet with project members during weekends or holidays. In addition, when the activities require certain expert competence and skills active participation is difficult or even impossible. Perhaps Spradley's (1980) term "Moderate participation" can best describe my degree of involvement since I participated in some activities actively and some activities more passively, which we will come back to. I did not become a project member. Nevertheless, as mentioned before, the fieldwork went on for a year, except for the time when the project members were on summer vacation and when I went back to the university to teach and attend a couple of conferences. Ideally according to Moeran (2007) the duration of fieldwork should be about a year.

Agar (1996 p.160-161) suggested keeping "informal interviews as the core of ethnography, with observations in a supplemental role". Merely observing actions without knowing the meaning that those actions imply for those involved is not what ethnography is about, he argued. Geertz' (1973) famous "wink" example shows that one and the same social situation may be interpreted in different but equally logical ways. What is essential is what the situation means to the people of the situation. "A major way to learn those meanings, especially in the early stages of ethnography, is to ask people what they are about", Agar stated (1996 p.160). Alvesson and Kärreman (2011) proposed to use the interview as method primarily as a "mini-seminar for idea-generation" (Alvesson and Kärreman 2011 p. 105) which I will come back to later on in the process when I discuss my participation in the project evaluation.

Moeran (2007 p.118) on the other hand, favoured observation and discussed "impression management" and the problematic situation that researchers who use interviews as their principal technique face as they try to "distinguish between what people say they do and what they actually do". He compared the interview and the participant observation situation and said about interviews that "people are always trying to manage impressions and to put across an image that may in fact be rather different from that of their "real" selves (…) "It becomes less so when that same interviewer had been hanging around the office for the past three months, watching what is going on and asking questions of anyone who has the time or inclination to talk to her." (Moeran 2007 p.118). Alvesson and Kärreman (2011) also propose that observations can be used to identify and interpret practices and discourses in different social settings. Observations are time-consuming but an open window to study real-time ongoing action, resulting in material "rich in detail and meaning" from natural contexts as Alvesson and Kärreman (2011 p. 108) admit.

Interviews and observations can thus be used in fruitful combinations; for example, confronting observations with interview statements to find inconsistencies or ambiguities, or conduct interviews to get background knowledge to be able to make sense of social interactions and identify patterns of meaning through observations (Alvesson and Kärreman 2011), which corresponds quite well with my experiences and how I used interviews. Agar (1996) discussed also how to blend techniques and argued that the field notes or "working notes" (Agar 1996 p.162) should include ideas from observations to follow up with interviews or the other way around, interviews provide ideas what to observe and record in the field notes.

Field notes or diaries were as mentioned mostly used to process or digest information in my case. Observations were a direct entrance for me to the ongoing process as it happened in real-time. Informal talk helped me clarify observations. Without the possibility to talk to people in between meetings in the corridor or by the desks and during lunches I would probably have had serious problems making sense of the communication and interaction and the project on the whole, due to its complexity. In addition, people talking informally, freely and spontaneously provided a natural setting compared to the talk produced in a well-prepared meeting (or interview).

Especially in the beginning project documents were needed to bring order and structure in what was going on around me. Much information about the project's official objectives and organization was to be found in different documents and in the beginning of my fieldwork I could better consult the documents for answering such questions than take up the project members' precious time to answer everything. However, documents are at best condensed versions of reality, sometimes they did not seem to correspond to what was going on at all. In addition, they often constituted the rational technical side of the project and the informal talk and the interviews the “emotional” or personal perspective of the project, the interviews and the small talk were thus needed to get the human side of the project.

Last, active participation was not possible in the beginning but constituted a research technique later on in the process, as I will explain later in this chapter. All in all, the different techniques yielded different “data” that could strengthen, complement and contrast each other. It helped me identify differences in project members’ experiences and viewpoints. These research methods provided with material out of which I generated different interpretations to “test” and mentally play around with and so in the end create comprehensive understanding as will be more explained and demonstrated later on in this chapter.

3.2 In the middle of the process

In the middle of the process my involvement in the field intensified in several ways. This is explained in section 3.2.1 “Intensifying involvement”. Social immersion, as ethnographers commonly call it (Spradley 1980; Prasad 1997), brings research advantages since it facilitates deep exploration and understanding from the “natives’” point of view. Yet, it came with a certain dilemma and problems of ethical character which also needed to be considered. This is explained in section 3.2.2 “Being an insider and outsider at the same time”.

3.2.1 Intensifying involvement

In the middle of the process, approximately from March 2004, the fieldwork changed character. It started when I came back to the project after I had been teaching at Linköping University (January - February 2004). I got more involved and became some sort of a group member even if I did not do the same work as the other group members. I formed my “own role” in the project and at PPM. This will be explained in this section.

The first thing that happened when I came back from the University was that I gave a presentation of my work up to this point for the PPM board. It entailed a summary and reflection of what I had “seen” and was a way for me to give feedback to my “host” at the same time as I got (short) feedback or comments from them on how I had understood and interpreted the project. Some of them were surprised that I had got such a detailed understanding of this complex project in such a short time. One of the subproject managers suggested that I should convene with them and give the same presentation for them. This was arranged more like a seminar and included both a presentation and discussion. It seemed that they enjoyed discussing and reflecting over the project work and the project groups, which probably constituted the main benefit for the project managers. For me I learned more details of different things, for example the internal relationships within groups, current and old conflicts and problems concerning organizational, communication and competence issues, discrepancy between consultant driven projects and agency work principles. As some of the early interview occasions described above, this turned out to be as a mini-seminar (Alvesson and Kärreman 2011) where the seminar participants aided the creative process by coming up with different interpretations and analysis contributions. Equally important, I also got to know the participants of the seminar better, that is, the different subproject managers, in terms of leadership style, their role and relationship in the project and in the host organization, work issues, personal views of the project and its members. They wondered what I planned to do with all information and I tried to be as clear as I could even though I did not know yet what I would write about in the end. I said that I wanted in the first place to learn what and how this project was managed and that I was curious about how people with different expertise areas are able to understand and communicate and collaborate with each other. However, since I in early phases thought that I would focus more on interaction and communication I had to be careful and beware that I did not reveal such things that could influence how they interacted or spoke with each other. These presentations and seminar were very valuable in my learning process and seemed also to help me build even more confidence and acceptance.

The second thing that occurred was that I moved from "my" office at the top floor to the open landscape two floors down where the designers, developers and testers and key project people worked, so that I could better follow the process and project members' everyday talk. I got a chance to listen and learn from them as they were working, which complemented the observation of meetings in a good way. It became also easier for me to hear about upcoming activities and different plans so that I also could arrange my agenda and plan forward in a better way. Since I often heard when the plans were being made I could also ask right away if I could join. Rather quickly it resulted in that I was often asked if I wanted to join before I had asked. As time went by, I easily would have been able to actively participate a little bit in the meetings, for example, help them recall things from previous meetings, see alternative solutions, take part in discussions regarding organizational things, and I sometimes heard when people misunderstood each other since I often had listened to the same conversation before from another perspective or with other people, I sometimes knew where a missing person was occupied with but I kept quiet and did not intrude since I as a researcher did not want to "contaminate" the situation or the outcome more than I already did with my mere existence. I also wanted to keep a certain distance to the project to be able to critically analyze what occurred and also be able to interpret it from different perspectives, that is, to avoid coming too close since that might make it difficult to generate different interpretations. "Detached involvement" (Agar 1996) constitutes a balancing act of on the one hand immerse in the social setting and learn all its complexities from the natives' point of view and on the other hand keep the distance in order to work as a researcher and critically investigate what goes on in the setting. Through detached involvement the studied culture can be mirrored, expressed and interpreted in terms of the researcher's culture (Garsten 2004, Van Maanen 1988).

Beside various meetings, I participated in (observed) workshops and seminars, which most of them took place at PPM, in the house. However, once I joined some project members/line personnel on a day conference at a conference center in Stockholm. The reason to take a day and sit somewhere else was to be able to work intensively together without being distracted by other obligations, people, messages, and from a research point of view this occasion was like an intensive course where I learned deeply both about the task and also how they worked together. Also, during this period, in the middle of the research process, I travelled with some of the project members to the Söderhamn branch twice; first by train and the second time by a chartered bus. The travel itself, the commuting time, the lunch and coffee breaks to which I was invited was valuable since I during these hours really had the chance to talk to people in a more personal manner and in this way strengthen my affiliation with the project and its people. I participated in some system tests and education activities (people worked in small teams during these occasions) and got a chance to see the user side of the system, which complemented the technical discussions, drawings, and details.

In the beginning it was hard to participate due to lack of understanding, in the middle process it was hard not to participate due to the involvement that I had been working up during several months. By way of anecdote, I give two examples of my influence on the social setting in which I "lived". The first concerns the board meeting. At the project board meetings I usually sat down in the same chair by the same side of the conference table every week until one day when I arrived and realized that one of the department managers/board committee members had entered the board room before me and taken "my" seat. I did not know where to sit and hesitated for a short moment and started to look around for a new place to sit. This manager/member noticed this and got also little confused and said "I'm sorry, did I take your seat now?" "No, not at all". "Oh yes, I sit here instead" he said and left his chair for me. It was quite embarrassing but we joked and laughed together about this peculiar situation.

The second story concerns the IT-meeting involvement. I almost ended up as a regular member of the recurrent weekly meetings and I had the meeting schedule "programmed" in my head. Since I did not want to be late to the meetings I usually began to pick up my stuff for the meetings before the other members and the Head Project Manager. This developed into a spontaneous "routine" where the Head Project Manager noticed when I started to pick up my recorder, pens and note books and began to use my meeting preparation as an alarm signal that it soon was time to go to the meeting. I was so involved in the meetings in the middle of the process people missed me if I did not show up. Once when I came back to the project from Linköping University, I had been away for some weeks, several project members had missed me and wondered where I had been. I had only told the Head Project Manager since I did not realize that the project members cared about my presence. Some people said I had missed interesting things (for me, they argued) and should have been there. Once, some developers and designers even faked a meeting and a conflict for me, that is, they set up a meeting time and said I was welcome to attend since they had something interesting to discuss. However it was all made up and the problem did not even exist. They fooled me with a practical joke when I came back!

The project managers and the developers had a good sense of humour and we often laughed together. Some of the project members also wondered every once in a while what I saw, what I was thinking, how the research process went on, which was important to convey in a way that made sense to them in order to uphold trust and acceptance, as is also discussed by various ethnographers, e.g. Agar (1996) and Spradley (1980) and Prasad (1997). Some of the project managers and I happened also to discuss different aspects of life in general, not only those related to the project. It seemed to me that a good relationship or *rapport* (Agar 1996) had been established.

To conclude this section in ethnographic terminology, I may describe this period as a time where I deepened my understanding through in-depth investigation and focused observations (Spradley 1980) of the different parts and activities or "domains" (Spradley 1980) that the project consisted of. Now I also searched for more explanations, relationships and patterns compared to the first months which mostly consisted of "descriptive observations", using Spradley's (1980) terminology. The "grand tour observations" using a "wide-angle lens" (Spradley 1980) in the beginning that aimed at achieving general understanding, although it also included details and focus on specific occurrences, were now complemented with "mini-tour observations" to more fully understand each stage in the project's development process, the technical aspects of the system, all different activities within each step, each project team and the relationships within a team and between the teams and the line organization.

3.2.2 Being insider and outsider at the same time

The intensified involvement and closer relationship brought me into an ethnographic dilemma of being an insider and outsider at the same time, that is, being a member of the group and then report on the same group of people. I was not there on behalf of someone else; I was not a "spy" who pretended to be a member and then in the end plan to use the information to harm the "informants" but I would still write about my "new friends" which at all times implied that they were reported on in public. I was also an outsider in the sense that even though I got to know some people better than others I could not take part for anyone. I tried to stay "neutral". I was there as a "Professional Stranger" (Agar 1996) with ambitions to learn the "the PPM and project members' "way of life" by living with or among them. But I had different purposes in mind and different responsibilities and obligations than the project members, which implied that I did not become one of them "for real". This problem is widely discussed in ethnographic readings (e.g. Agar 1996, Spradley 1980, Prasad 1997). It is normally associated with ethical issues that the researcher is ought to consider. I will show how I dealt with this using Spradley's (1980 p. 20) discussion on ethical principles.

First, I was careful about the information I heard and read in order to "Safeguard Informants' Rights, Interests, and Sensitivities". Personal conditions and relationships have been excluded from the written report. I "Considered informants first" in the sense that I would not let my informants down or reveal harmful information about them to managers or other potential interest groups. Furthermore, as has been mentioned previously in this chapter, I have tried to "Communicate Research Objectives" and intentions as clear as possible but since I entered their community with an open research approach I could not explicate exactly what the objectives or research questions were but explained the research interest in general and continued the conversation about the goals and intentions and what I found interesting throughout the process. I also "Protected the privacy of Informants" in the sense that project members could choose to be anonymous in the written account. This study did not either focus on social or personal sensitive issues. Moreover, according to Spradley (1980) and Prasad (1997) one should not "Exploit Informants", that is, the "informants" should also gain something from the research, not only the researcher, and get a fair return. As mentioned before, I presented the research twice and participated in a seminar in the middle of the process which I believe contained some feedback with new insights and understanding for them. I also participated in an interview in the internal newsletter and on the PPM intranet but most importantly, I was deeply involved in the project closure and evaluation, which will be explained in the next section. Beside two oral presentations I also wrote up my interpretations and comments in a special report for PPM and the project. The thesis itself is of course also accessible to whoever is interested so in this way I also "Made Reports Available to Informants", which made up Spradley's last ethical principle.

3.3 Before walking out

Before I finished the empirical study and went back from the field to the University I assisted the project in the project's evaluation process. This was a good way for me to give something back to the project at the same time as I extended and enriched my material further. I conducted over 50 interviews that lasted for about 1 up to 2.5 hours each. My participation in the evaluation is described in section 3.3.1. The following section 3.3.2 involves a description of additional evaluation activities that together with the main evaluation activity constitute what I call "fair return". The section contains also a discussion on how both the project and I gained from these activities in some sort of "reciprocation".

3.3.1 Participation in project evaluation

According to the principles written in PPM's project handbook the project should be evaluated at the end of the process. The IT department's traditional way of doing this was through group seminars lead by the IT department's project and quality coordinator and controller. The Quality Controller does not normally participate in the projects but brings instead an outside perspective in the compilation and analysis of the projects. When I heard about the evaluation procedure I thought that I might be able to complement the assessment of the project by conducting individual interviews with the project members. This would give me a chance to check my understanding and learn about remaining ambiguities, questions and relationships. Having deep-going reflection discussions with different organizational members would augment my material and complement the observations of on-going activities further. The Head Project Manager accepted the offer and most project members seemed to enjoy being interviewed and stated that they believed in this kind of evaluation since they in this way got a chance to explain their opinion, suggestions, perspective and ideas for the future in detail with someone who had followed the process closely and could discuss different events and situations.

The interview procedure can perhaps most easily be described in a "who, when, what, where, how–form". *Who.* The Head Project Manager gave me a list of people and suggested that I should interview at least one person from each project group and also representatives from the line organization including department managers since the project and its outcome affected the whole PPM organization. Beside this request I was free to do as I found appropriate. I gladly and gratefully interviewed the suggested people and added several members in each project group, including group leaders and project managers and board committee members, to take advantage of this opportunity and maximize the outcome. I interviewed people I knew well as well as people whose voice I hardly had heard but whose skills and work I recognized and wanted to learn more about. I was satisfied with my sample since almost all "key informants" and primary people were included and all different groups represented.

Some people that I interviewed offered more exciting viewpoints than I had anticipated so I was content that I did not only choose to interview people I had observed and heard of before. I enlarged my view and got a more holistic understanding of PPM during the interview process thanks to the discussions with the "unknown" people. Agar (1996 p.168) stated that the researcher should include a wide range of informants; "If you only check what you have learned against the people who taught you, there is a good chance of success, unless you have truly misunderstood. But if you check what you have learned among that group *and* among others who have not talked to you that much, you build the credibility of your statements as representative of the entire group."

Almost all interviewees were enthusiastic about participating and argued that evaluation was important. One refused my invitation though and simply said no without explanation. This person's group manager explained however that it is common that developers are only interested in technical things and do not want to engage in other activities. There were nonetheless other technicians and developers that I could interview which reduced the "damage" of this loss. It happened a couple of times that people that I had not interviewed came to me and wondered why I had not asked them and said that they wanted to be interviewed and participate in the evaluation process. I wanted to interview as many as I could so I greeted the requests. In total, this interview session comprised over 50 interviews as mentioned above. Most often I conducted individual face-to-face interviews but there were also two occasions where I met with two project members at the time and another two interviews that were carried out over the phone.

When. The interviews were carried out during and after project closure in October and November 2004. *What.* The only thing that the Head Project Manager explicitly asked me to investigate was whether there was a conflict between the line organization and the project. The background was that there had been a rumour that the project had been in the way for different line activities and hindered other plans and projects. Except for this I was free to design the interviews as I considered appropriate. The interview material should help me generate detailed descriptions of the project process, since I would not be able to attain credibility in the research community or at PPM if I did not have enough descriptive details to explain what had occurred, but the interview had more purposes than to describe the reality out there; the outcome of the interviews should help project and organizational members reflect and perhaps improve the business in the future. The interviews should also generate research material containing different perspectives out of which different interpretations could be made in the research text. *Where.* I was assisted by the project and PPM administration with booking a small conference room at PPM where the interviews could take place. Project and organizational members were asked to sign up for an interview in the interview schedule, also at least partly organized by the administration at PPM.

TIME DIMENSION	**Requirement process**	**Design and Programming process**	**Test process**	**Whole project process**
Individual				
Group				
Project management/Board/Administration				
Whole project organization				**SUM/Goal achivement**

Table 2: Interview matrix

How. I needed to create an interview design that could serve both research and project purposes. The material from the interviews should permit interesting research discussions but also allow for concrete analysis and conclusion to be presented to the project and organizational members. Before carrying out the interviews I designed something that I called "Interview Matrix" illustrated in the figure above. The matrix contained the different subproject groups and processes (requirement process, design and development and test process) on one of the axes and the organizational level on the other (individual level, group level, management level). And each box in the matrix was filled with what turned up during the interview. The bottom line considered the project as a whole and the right column constituted the project process as a whole. The third dimension related to time, which was an integral aspect throughout the project process and integrated in each square of the matrix. After all had been gone through the essence was summarized in a final note in the box down to the right with a comment on whether the project was successful and had achieved its goal.

The matrix was used during the interview as an interview guideline. It denoted or framed at an overall level what the interviews should cover but it did not specify exact questions to be asked in a linear mode. I wanted to keep the interview "half-open" or semi-structured to be able to have a conversation or discussion with the interviewees that could capture unexpected issues and spontaneous questions and answers. At the same time I wanted to be sure that I had gone through the same topics with all interviewees. Since this was also a project evaluation activity I reasoned that it was important that the participants had been given the same opportunity to discuss and comment the same things in the same way so that they did not need to wonder if I had treated the interviewees differently. Yet, since they worked at different organizational levels, in different groups and had different relationships and perspectives on the project the questions and discussion subjects could not be too narrow but should rather differ a lot.

I started the interview by presenting the matrix and said that it constituted an overall frame or checklist to ensure that we in the end of the interview had discussed all different aspects of the project. On paper it might have seemed like a firm structure. I explained it was more of a general outline from which we would develop an informal conversation taking different turns or loops rather than a firm schema with a strict pre-determined linearly order. After presenting I asked if it was okay and if they had other expectations of the session. Everyone agreed and accepted the idea. Some were a little worried that they did not have so much to say but I ensured it did not matter; everyone answers what he or she knows and wants and so tells the story from his or her individual perspective. I inspired the interviewees to take the opportunity and use the interview as an occasion to shell out whatever opinion and whatever they have had in mind regarding the work situation and the project during the past two years. I strived for achieving a feeling that the interview procedure was for them, since it actually was for them, but at the same time I had my research interest and commitment in mind. I got a chance to explore details of events that I had observed during the process, paraphrase to make sure I had understood different things and practice active listening to come close the interviewees' "worldview" in an informal and personal way.

However, most of all the interviews constituted "mini-seminars for idea generation", expressed in Alvesson and Kärreman's (2011) terminology. People were full of ideas, insights, reflections and opinions that they wanted to share and discuss with me. I believe that organizational and project members seldom have/take the time to sit down and reflect over their work situation. In the interview situation they could talk for an hour about different experiences, ideas, and improvement suggestions. Some of the interviewees seemed also interested in teaching me about different things and help me out in my research process. I asked about unexpected or surprising outcome and events and deviations from plans. These were descriptive but also analytical interviews (Alvesson and Kärreman 2011) in the sense that we in conversation did not just only describe things as they were trying to mirror reality but also question or re-think different events and relationships. As a result, it always ended up in a positive way – people had much to discuss and some were even surprised that they were able to talk for such a long time; "I did not believe that I would have things to say for a whole hour" as one of the interviewees commented. It seemed that if the interviewee knew the interview plan and structure and what was to come, he or she could relax; there would be no bad or uncomfortable surprises. To be open and clear with the interview design appeared to build and strengthen trust and acceptance.

3.3.2 Fair return and reciprocation

The result of my evaluation and the outcome of the evaluation that the IT department conducted were presented at the project's closure and celebration day in January 2005. I was also asked to present the evaluation for the PPM board at a board meeting some weeks later. I also handed in a 20-pages report especially written for PPM. PPM used this report as a basis for a more condensed non-academic PPM paper or PM where concrete organizational changes were shortly suggested to be implemented in the organization. In addition, I was asked to participate in the internal newsletter. I was interviewed for approximately an hour about my experiences from observing the project and how I had interpreted the outcome. The evaluation process presentations and report, together with earlier presentations and seminars can be seen as payback to my field host. The project members and management and I never had a discussion about what would constitute a "fair return" (Spradley 1980) but as I understood the comments from various project members it exceeded expectations and both project members and some research colleagues wondered how much I was paid in monetary terms (which I of course was not at all). Referring to Prasad (1997) I argued that if I shared my findings and interpretations in different ways with the project members, they would feel less exploited when I then wrote about them in the thesis and in various research papers.

Furthermore, the evaluation process constituted also a way for me to participate and reinforce my involvement even further in their project activities at the same time as giving something in return. In addition, taking part in this activity gave me also a chance to check my understanding through paraphrasing, which according to Agar (1996 p.128) is a "powerful test of comprehension" and listen to the project and the organizational members' reaction to my reflections and interpretations and also see if I had missed any important events or relationships as argued before. Project members' view and comments on my reports and presentations were mostly positive and affirmative and people seemed to recognize themselves in the story, and verified that my interpretations of what had occurred were possible, which according to Agar (1996 p.131) is important in the process of giving an account of a foreign culture. Some people appreciated the more "academic" interpretations and connections among things and commented that it constituted ways of seeing that they had not been thinking of in the same way before.

The evaluation work also meant that I processed my field material and "kneaded" it intensively in different forms. The processing of "data" started the first day in the field and continued long time after coming home but these specific evaluation performances (oral presentations, discussions, written reports, internal newsletter participation) intensified the work with the material and pushed the process forward since I was "forced" to organize my thinking and come up with something comprehensible that could be presented to other people.

All in all, I suppose that the evaluation process with its different activities can be described and characterized with the word reciprocation since it seemed to produce winners but no losers. I stayed in touch with some of the project managers and was so updated on what was going on in the organization during the following years. In 2006 when I revisited PPM and the IT department they informally presented the changes made in the development method, which basically constituted a shift from running long and big projects to using a scrum-similar development method, and suggested that I should start another study to see how that worked because it was even more exciting and much better, they argued. It was tempting but the dissertation could not contain more than one study and not my head either since it already worked hard to process the experiences (including 400 hours of conversation, almost 60 interviews, hundreds of pages of documents) made from the first study. Nevertheless, the processing of the empirical material continued at home after the evaluation procedure and this is discussed in the next section.

4. Back home

I started the fieldwork as a grey eminence, continued as an insider and returned detached again after the participation in the evaluation process was completed. As Agar (1996 p.251) discussed doing ethnography involves some important contradictions that the researcher must deal with in some way; “humanity and science, involvement and detachment, breadth and depth, subordination and dominance, friend and stranger”. Agar’s approach to handle this is in line with my experience “the first member of each opposition is emphasized in the beginning, the second member at the end”. In this last section of the chapter we will first see how the processing of material continued; in 4.1.1 we look into how the empirical material base was generated, and in 4.1.2 it is described how the material was delimited to be developed into a research account. Section 4.2 contains two subparts. First the writing endeavour is explained (4.2.1) and then some method reflections are presented (4.2.2).

4.1 Continuing the processing of material

The first part of this section, 4.1.1, discusses how the information and experiences from the field were constructed and formed into an encompassing empirical base. Under the next subtitle, 4.1.2, I explain the “data reduction strategy” that I used to delimit and select what to focus on and include in the empirical account of the thesis. Before going into these descriptions I shortly summarize the field data in numerical terms and make a comment of the abundance of data.

In the end I had approximately 200 digital files consisting of meetings and interviews (almost 60 interviews) that spanned from 20 minutes up to 3 hours each - in total almost 400 hours of talk. In addition to this I brought material and experiences from many meetings and carried out informal interviews and conversations that were not recorded. I had also hundreds of pages of site documents in my suitcase and 4 personal diaries and of course the already elaborated material and drafts that had been written and orally presented during the process. Data abounded.

However, I peculiarly enough did not feel that I drowned or was overloaded by data per se, only in ideas what to do with it, how to interpret it and what image to project. I learned from inside and therefore I never had to fight with memory problems. I learned in the same way as one learns a new job - all my senses where activated to experience and come close and take things near my heart. This process involved much contextual learning of information systems and the Swedish pension system the IT consulting industry, the public agency world, and the contrast between the two “planets”, project management, which also resulted in that I did not need to fight strongly for remembering things out of head, instead the cultural scene, the actors, the drama, the problems, the backdrops and the props became a part of me as well. It would not have been the same if I only had conducted interviews or laboratory observations detached from the context, the milieu, the other people, the physical setting, the smell, the atmosphere, the jokes, the scenes, the text and pictures on the whiteboards. Then I would have had to work up an improved memory to be able to remember what the participants said and did. I have been living with this every day for several years and kept it alive through constantly be thinking of the people involved, listening to their voices and writing about what happened. Also, during the years I have constantly worked with the material and rewritten the text innumerable times. Thanks to the recorder and the computer things, places and people never become “history”; everything is in a steady motion, in a constant “now”.

4.1.1 Generating the empirical base

The processing of data, that is, interpretation and sense-making of what was seen and heard is indeed not something that starts when the empirical field study is ended. It is rather something that commences immediately when entering the field and continues throughout the process until the thesis has got its final version. Yet, occasions such as the evaluation activity, the presentations, feedback reports and reflection seminars, written progress reports, interviews. intensify the elaboration with sorting, organizing and categorizing the material and enhance reflection and interpretation. I had thus processed "data" or generated empirical material (Alvesson and Kärreman 2011) while working on the feedback given to the organization and the project, as previously explained. The material was also processed and created when I analysed and wrote shorter versions and selected pieces of the study to research conferences, seminars and PhD courses.

What one pays attention to and actually covers in the field and how things are interpreted depends on the researcher's personal and cultural background, and interests of the researcher and the research community, as is widely discussed in social sciences and qualitative method literature (e.g. Agar 1996, Alvesson and Kärreman 2011, Garsten 2004) and mentioned earlier in this chapter. Agar (1996 p. 98) stated "Whether it is your personality, your rules of interaction, your cultural bias toward significant topics, your professional training, or something else, you do not go into the field as a passive recorder of objective data. During fieldwork, you are surrounded by a multitude of noises and activities. As you choose what to attend to and how to interpret it, mental doors slam shut on the alternatives." The researcher is perhaps aware of some influencing factors of the background while other biases certainly are hidden (Agar 1996, Alvesson and Kärreman 2011). Even though the personal and social background and knowledge is important to be able to "detect" things at all, it can also carry some problems. For instance the researcher may unconsciously see and interpret experiences in a way that confirms pre-decided thoughts and ideas (Alvesson and Kärreman 2011). Similarly, as Agar (1996) discussed, ethnographies sometimes reveal more about the researcher than the observed social group. The researcher may unconsciously use the study and the writing to discuss and treat own problems instead of deeply going into the different natives' perspectives. The study would then not be as exploratory as it perhaps was stated and intended to be.

However, there are different strategies to handle this. For instance, I was careful in making sure I had learned and included different people's perspectives. This is a prerequisite to accomplish informed "thick description" (Geertz 1973). I also let different people's voices be represented in the text on equal terms so that either group was attached certain attributes before I had really learned about them. Prasad (1997) asserted that in (studies of) systems development, less powerful groups such as data entry operators and those who resist technological change are often regarded implicitly in a pejorative way as less knowledgeable or old-fashioned. The researcher should beware of such tendencies and reflect over how such groups are represented, why resistance it treated depreciatively and what space such voices are given in the written account.

Agar (1996) suggests that the researcher should document and show the methods used and how the conclusion was reached, (which basically constitutes the essence of the method chapter) question assumptions and interpretations and describe as much of ones biases as possible. For this end I used different theories as explained before to see the material from different angles to avoid being steered of pre-determined literature and given assumptions and achieve alternative interpretations (Alvesson and Kärreman 2011, Agar 1996). The theoretical-empirical interplay was a natural part of the process through paper writings, PhD-courses, research conferences and also from daily writing-reading process. I also asked for feedback from the project workers and organizational members about my interpretations and understandings to check that things meant to the field members what I said it meant to them, as Agar (1996) discussed and make sure they could recognize themselves in the representation, even if they must not agree with all interpretations, as Prasad (1997) stated.

In addition I have had many informal conversations as well as formal discussions in research seminars with research colleagues to test different ideas and connect interpretations to other works in the same area and so elevate quality of research undertakings, which is a pragmatic way normally used in research community to reduce personal bias (Alvesson and Kärreman 2011). Taken together, these constituted some strategies used to deal with the personal background and biases that I might have brought into the interpretation of empirical material.

I elaborated the empirical account through reading my notes and field documents, listening and also partly transcribing meetings, interviews and different conversations. I zoomed in on different events and analyzed details with a microscope perspective and I zoomed out to chart the overall project process from a helicopter perspective. This was done over and over again. The best way for me to work up the empirical base was however through writing. In some papers I wrote summaries or short versions, in other papers I selected aspects to focus on and analyze from certain perspective. Much thinking occurred in the process of writing. When things were to be expressed and put down on paper I could see their relationships, identify the inherent complexities and interconnections, detect inconsistencies and interesting problems, sort among events and essentials. Themes and processes as well as important elements or components evolved during the writing effort. Similar experiences and way of working are described in for example Spradley (1980), Agar (1996), and Van Maanen (1988) and as Wolcott (1990 p.20-21) said *"You cannot begin writing early enough."* and "writing is a form of thinking" (…) Writing is a great way to discover what we are thinking, as well as discover gaps in our thinking".

The basic empirical account that I elaborated, depicted in the illustration below, entailed description of the different processes of the project; the requirement, the design and programming, the testing, the overall project process and the management or leadership process, depicted by the horizontal arrows in the figure below. It also contained theme descriptions, represented by the vertical arrows, such as collaboration, communication, interaction, routines, tools and critical events of the different processes. Each intersection point is covered in the basic empirical description with detailed component or activity analysis. In addition, when searching and writing about current topics more or other types of themes and processes constantly evolved, which implied that the material was never locked or fixed into specific categories but left open to constantly be reconsidered as I learned more by reading and listening and analyzing through different theoretical lenses over and over again. This is represented by the boxes "Emerging themes" and "Other evolving processes" in the illustration. None of the categories or processes, except for the overall project process were set out and specified from the beginning of the study.

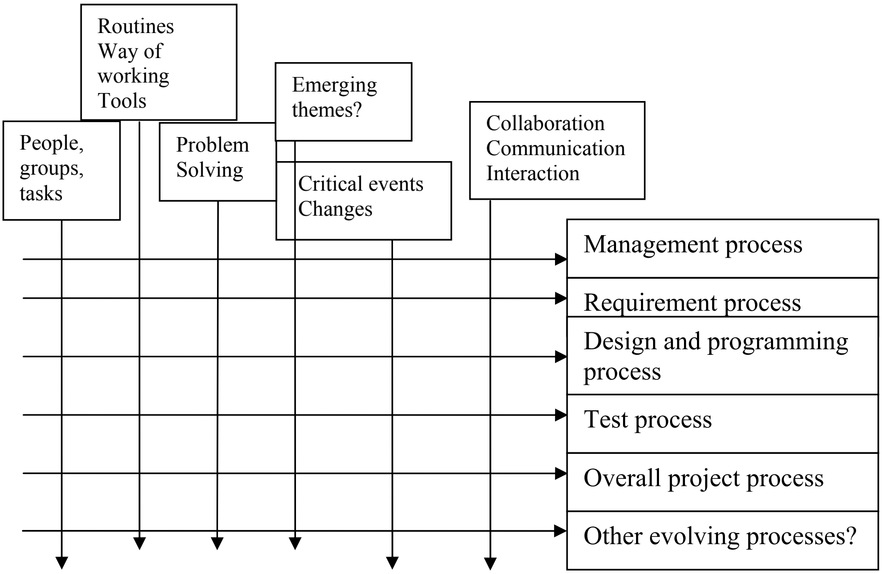


Figure 3: Generating the empirical base

During the first year after I got back home I saw new things every time I went through the material. Once I opened the notebook or a computer file or “asked” the empirical material something I got new answers and detected new things that offered new threads to investigate and reflect over. Some themes and process events persisted of course but nuances shifted for a long time and details that first seemed irrelevant suddenly became super-important, and vice verse. I described the processes one at the time, shifted to describe the different themes one at the time across the different processes. I tried to understand the different project members’ ordinary workday and all the project groups’ various routines. I also described critical events and changes or turning points and unusual and potentially extraordinary interesting things. I inquired into different dimensions of the project, such as the technical, the social, the organizational, the symbolic and the historical dimension. I went into the different aspects taking different grips; staying close and near the project workers’ point of view and opinions through transcribing and staying close to quotations and contrasted with empirical distance and questioning whether there was anything new or interesting in it from a research point of view. In this way I twisted and turned the material to see it from different angles. Plain descriptions as well as contrasting elaborations constituted a way for me to generate a comprehensive account, out of which I then could choose what to include in the final draft of the thesis and what to dismiss. This takes us to the next section.

4.1.2 From all-inclusive to highly-selective

One well-known problem with this type of research method is that it commonly yields much more material than can be included in a thesis (or perhaps in any writing). As a result, one of the main challenges that the researcher faces is to decide what to include and what to leave out. As Genzuk (2003 p.9) put it “The agony of omitting in the part of the researcher is matched only by the readers' agony in having to read those things that were not omitted, but should have been." As I was processing the material and reflecting over my experiences the empirical base grew and could probably have resulted in hundreds of pages. How did I go about delimiting the material and selecting a topical focus?

I first identified all events and problems that seemed to be extraordinary important to the project participants and "plotted" them sequentially in accordance with the project's time line. Then I choose to focus on those episodes, problems and solutions that could be related to my research interests, that is, to interdisciplinary collaboration, communication and knowledge integration. Other events, for instance summer vacation planning problems, media and systems version release planning troubles, and detailed technical solutions decisions and similar were omitted or at least downplayed in the empirical narrative.

Inspired by Spradley's (1980 p.105-106) descriptions for selecting ethnographic focus one can thus say that the criteria consisted of "suggestions from informants" and important episodes from the informants' point of view, "theoretical/personal interest", and identification of the most prominent "organizing domain", which here was the waterfall-like sequential project process design. It constituted the *main organizing domain* and work principle that connected and kept different sub-processes and relationships together. I reasoned that the project process was an appropriate text organizing principle since it comprised different steps that were conducted successively and chronologically one at the time, at least at an overall project level (within and between steps there were some bouncing back and forth), which also could match a normal linear way of reading. The project's development process constitutes thus the "backbone" (Spradley 1980 p.158) of the thesis whereas the collaboration and knowledge integration-related problem solving situations became the "universal theme" and topical focus throughout the process (Spradley 1980 p. 153). As this had been chosen and settled I could also develop and formulate the research questions and the purpose more precisely.

4.2 Writing up and presenting the project story

As the empirical material had been worked up and a focus selected it remained to translate the material into a legible and interesting account. The essence of this process is to represent a culture in terms of another, as ethnographers often state (e.g. Agar 1996, Van Maanen 1988, Garsten 2004), which means that the researcher first decodes the meaning system of the culture under study and then translates and encodes it into the language and meaning system that is used by the target audience (who typically belongs to another community or culture). Writing ethnography requires hence deep knowledge of two cultural meaning systems (Spradley 1980, Agar 1996, Van Maanen 1980, Geertz 1973). Organizing and analyzing empirical material of ethnographic nature takes time and is quite demanding. However, the most challenging task is the writing process. As Van Maanen (1988 p. 7) said "Culture is not strictly speaking a scientific object, but is created, as is the reader's view of it, by the active construction of a text." The experiences from the field and the empirical material is not inside one's head to be "uncorked like a bottle and a message poured out" as Van Maanen (1988 p.xii) put it. It is an intellectual workout requiring blood, sweat and tears. In 4.2.1 it is explained how the material was divided, presented and interpreted using different levels of abstractions. In 4.2.2 some reflections on the choice of method are discussed.

4.2.1 Analysis, interpretation and synthesis

"An ethnography should be empirical enough to be credible and analytical enough to be interesting. Too little or too much of either is presumably deadly." (Van Maanen 1988 p. 29) This is a challenge that the researcher faces in the process of writing. I have dealt with this balancing act in the following way. First, in the same way as Genzuk (2003) discussed, I distinguished between analysis and interpretation. Analysis is a search for patterns (Spradley 1980), and offers an overview of the entire "picture" or process (Genzuk 2003). The analysis part contains close-to-the field-descriptions where "raw" materials in terms of quotations are frequent. "In order for a reader to see the lives of the people we study, we must show them through particulars, not merely talk about them in generalities" as Spradley (1980 p.162) put forward. I let the natives' voice be heard through the quotations and this was also a way for me to discipline my interpretations and imagination and secure that I did not "go wild" or completely lose empirical bearing and closeness in the interpretation later on. I elaborated the analysis by using quotations and conversation extracts to pick up interesting viewpoints and interpretations of the natives in a natural way and arranged the quotes thematically in accordance with the projects chronological timeline. The quotations formed a dialogue where the project members spoke to each other. The first versions contained too many quotations and too little comments. The problem with that was that it did not mean so much to anyone who had not participated in the field process. It became hard for an outsider to read and understand the text so gradually I added more and more description paragraphs with contextual details to explain to the reader how and why things occurred and what it meant to the project members.

As a next step the analysis descriptions were dressed up a little bit with my view and understanding. I added my perspective and meaning to the analysis in an act of interpretation. "Interpretation involves attaching meaning and significance to the analysis, explaining descriptive patterns, and looking for relationships and linkages among descriptive dimensions" as Genzuk (2003) suggested. In this part I showed and explained what I considered most important and interesting having the research interests in mind, that is, the chosen topical focus and main theme, explained above in section 4.1.2, rather than adhering only to the natives' point of view. During the case construction process the research purpose and questions were more elaborated and this influenced also what was included in the different empirical parts.

The third step in the process of writing up the empirical story was to put all things from the interpretation together in a synthesis and create a new condensed shape of the material. This served then as the basis for the development of the new notion or concept. The analysis step is thus the most concrete level with a low degree of abstraction and assemblage. Then as we move towards interpretation and synthesis the text gets increasingly more abstract and compounded. Quotations become fewer and the natives' voices are implicitly stated and referred to in more general terms. At the most abstract and general level, in concluding chapters, project participants' viewpoints are imbedded in their "pod", that is, in the new suggested concept, and compared with other researchers' theories and conceptions.

In similar way Spradley (1980) explains the "translation process", which in the first stage entails describing cultural behaviour and local meanings. Then the next step is communication of the interpretation of this particular culture. In order to succeed with the translation and communication process one should acknowledge different levels of writing and handle them wisely, which according to Spradley (1980 p.162) is "to begin with the particular, the concrete, specific events of everyday life". (…) "the researcher moves to more and more general statements about the culture. With the discovery of more general categories and cultural themes, the ethnographer begins to make comparisons with other cultures and make even more general statements about the culture studied."

A final note in this section regards the conceptual distinction made between analysis, the *emic* description from the insiders' or natives' point of view, and interpretation, the *etic* narrative from the outsider researcher's point of view (Agar 1996 p.20). It is not perfectly congruent with the ethnographic methodology and approach adopted in this research since the distinction states that the first part contains pure descriptions from the natives' point of view as objective representations of something "out there", which is not the whole true story. The empirical material was *constructed* in order to create an understandable and interesting case description through my selection and arrangement of quotes and complementary descriptions and explanations. The researcher interprets continually what happens during the fieldwork process, participates in the life world of the people under study (and becomes sometimes interviewed by the native's!) and then tries to create an interesting story based on constructed material, and does not stand outside to report something objective (Alvesson and Kärreman 2011). He or she does not report exactly and inclusively what happened in the field either. This implies for instance that voices have been muted (e.g. IT technicians at the line department, security developers in the line organization, board members), and perspectives (the line organization perspective, pension saver perspective, PPM board/top management and project board perspective for instance), and aspects (such as HRM, technical choices and solutions, cost and profitability) and methodological ideas (e.g. discourse analysis, conversation analysis) and tentative ideas and interpretations such as the project as a knowledge creating instance and the view of the project as an inquiry system have been considered but abandoned and taken away, as well as empirical scenes and events (e.g. technical environment problem, education and training issue, implementation and transition planning) that did not fit into the "storyline". In this way the separation between analysis and interpretation, the emic and the etic, is not perfectly correct. "A statement would almost always contain some assumptions about perception or intent on the part of the group members, but it would also be constructed by the ethnographer in terms of his own professional context and goals" as Agar (1996 p.239) explained.

Nevertheless, this three step (analysis, interpretation, synthesis) procedure constituted a way to make the processing and interpretation of empirical material as well as the development of the new concept as visible as possible. I tried to demonstrate how the empirical material was created and used. Showing how things have been interpreted and constructed is often a major challenge in qualitative research and is important if the reader should be convinced of the inherent potential of the empirical material to serve as a base from which new lines of theories can be developed or new concepts take shape. The reader must be given a chance to judge the value and trustworthiness of the contributions, that is, to explore whether the suggestions have sufficiently empirical ground and are not too speculative. A credible case should serve as material for questioning existing assumptions and theories (Alvesson and Kärreman 2011). The attempt to show how the material was generated and used also aimed at giving the reader a chance to criticize or identify alternative interpretations, which will be further discussed in the up-coming section.

4.2.2 Reflections on research approach and method

In retrospect, this was not probably the most time efficient way of doing research. It is hard to know when to stop, when you have heard enough or have enough material to contribute with something interesting, especially for an inexperienced researcher. Most of all however, I did not want to stop. I wanted to be there in the field, hear more, carry out more interviews, write about it in yet another way, reflect a little bit more, refine, revise, change, erase, start all over again. I was lucky that the project under study finally came to an end so that new information eventually stopped turning up.

There are several limits inherent in ethnography too such as the personal background and pre-text assumptions that the researcher brings into the study and the research community's traditions and more practical ones such as access and relationship issues, time and budget aspects as well as writing matters, as previously discussed (Van Maanen 1988). Also different fieldwork techniques or methods carry advantages as well as disadvantages and problems. For instance, interview situations and accounts may be infected by interviewees' hidden agendas, political interests, impression management and identity creation attempts, application of scripts, memory errors and fallacies, as Alvesson and Kärreman (2011) and Agar (1996) explained. Additionally, there are problems associated with observations as well, such as sense-making difficulties of complex interactions and the influence that the researcher may impose on the people under study which may induce that they behave abnormally (Alvesson and Kärreman 2011). The researcher then misses the point of observations (to see how people act and what they do in their natural context).

I believe though that the kind of end-of-process seminar-like interviews along with the start-up interviews conducted, direct personal involvement and close observations of meetings and interaction over a extensive period of time in the informants' home territory, informal spontaneous talk, participation in activities, document readings and different kinds of feedback and comprehension "tests", all together reduced the influence of each technique's disadvantages. Researchers have called for ethnographic approaches in business administration in general (Alvesson and Kärreman 2011) and knowledge integration in particular (Tell 2011). This constitutes a type of study that may complement other qualitative and quantitative studies in knowledge organization and integration since it entails a from-inside- learning- perspective and has an open exploratory character – in doing ethnography nothing is given on beforehand or "hammered on top" of the people under study (Agar 1996), which potentially produces a case that can open up alternative views and new notions. This kind of material seems hard to get from survey studies, hypothesis testing, isolated interviews or mere observations, statistical desk-top research, experimental laboratory studies or secondary data investigations.

Nevertheless, a recent incidence reminds me that there is almost always more than one way of doing things. My four years old daughter said when I taught how to make pancakes: "Mama, I see, you start with the wheat and salt and then you pour the milk into it and last you add the eggs." "Yes, that is correct, it is according to the recipe in this cook book, you see, honey", I answered. After a short moment of reflection she wisely informed me: "At Grandmother's place we always start with the eggs. And that becomes pancakes too, mama."

CHAPTER IV

IV. CREATING A PENSION INFORMATION SYSTEM

1. Introduction

If you are interested in the realm of human affairs in combination with adventure, intensive work and advanced technology you have come to the right place. This is a travelogue about how a crew of diverse people got together to create and organize a complex new “mini-world”.

1.1 Envisioning peak performance

On December the 2nd in 1998 one could read in the Swedish newspaper “Affärsvärlden“ (“Business World”) that the most important decision made by the Swedish parliament in 1998 was to implement a new pension system. A new administrative unit of government, named the Premium Pension Authority (PPM), was to be built up in order to administrate the (compulsory) savings that all Swedish income earners were (still are) obliged to place in pension mutual funds. This saving is called the “premium pension” and constitutes a 2,5% share of the 18,5% of the wage that individuals and employers pay in pension fees. The new pension law implied that pension savers should have the possibility to choose among several hundreds of mutual funds and whenever and unlimitedly switch funds. Pension savers should also be able to follow and monitor their individual fund and pension developments and participate in the decision on when and how to apply for and receive the pension disbursements. The premium pension was in the beginning only a general concept and political idea - an unknown “business” area that the members of PPM were supposed to outline and put into concrete form.

To enable the administration of the new pension the personnel at the new agency needed an IT-system. The development of the IT-system turned out to be a long and problematic process. It started with contracting an external IT-consultancy firm. After almost a year of problems, in 1999, this contract was terminated. From PPM’s perspective, it resulted in a bill for SEK 170 million and a “useless” computer system. The development of the IT-system was then taken over by PPM’s IT-department. A year later, in fall 2000, an elementary system version was ready to be implemented to let pension savers make their first fund investments.

Nevertheless, even though the computer system now was up running, in late 2000, more functionality was needed. New versions, updates and patches were developed and released continually. PPM aspired to be a contemporary “customer-friendly” organization, that is, a modern agency with services developed with a strong citizen and pension saver perspective. One part of PPM’s mission was to attract people and make it interesting to actively manage the premium pension. PPM tried to accomplish this through, among different means, broadcast advertisements with stars and famous people. Also, the nowadays well-known (in Sweden) “Orange Envelope” was borne and sent out to citizens and became soon a familiar letter in Swedish pension savers’ mailboxes. It contained information about the premium pension savings, growth rate figures and expected pension disbursements.

However, PPM considered that the most important way to trigger people’s interest and become “pension-saver friendly” was to provide easily accessible modern services based on high-tech IT-solutions. This was in accordance with the “24-hours agency” vision that blew as a wind of change over the authorities in Sweden around the end of the last century. At PPM this vision included electronic administration of pension issues, modern interactive communication via the web, fast personal service, simplified and less bureaucratic processes (compared to how authority issues had been handled historically) high accessibility and easily obtainable information about different pension and mutual fund "products" and services. These ideas led to the start-up of a new development process in the middle of 2002. Despite many problems along the way, the process ended successfully with a new IT-system 2.5 years later, that is, in the end of 2004. This work became the "grand opus" for many organizational members of PPM. The thesis reports on how this new information system came about and covers thus the time period from 2002 until 2004. The development work was organized in two projects that were running in parallel the first year, that is, from summer 2002 to springtime 2003. In May 2003 the two projects were merged into one project which was finished in the end of 2004. This is pictured in the timeline figure below. The first year, from the summer 2002 until autumn 2003, was studied retrospectively while the last year of the development process was followed in real-time. How the development process unfolded from 2002 until 2004 will be described carefully later on in the chapter. It is a story about dynamic interdisciplinary collaboration, organizational problems and imaginative solutions. Before going into the details of this story, the outlining and elaboration of the premium pension is described as a task involving exploration and creation of a new "world" (1.2). Then it is described how that in turn called for exploration and creation of new IT-technology (1.3). After this the travelogue can commence (1.4).

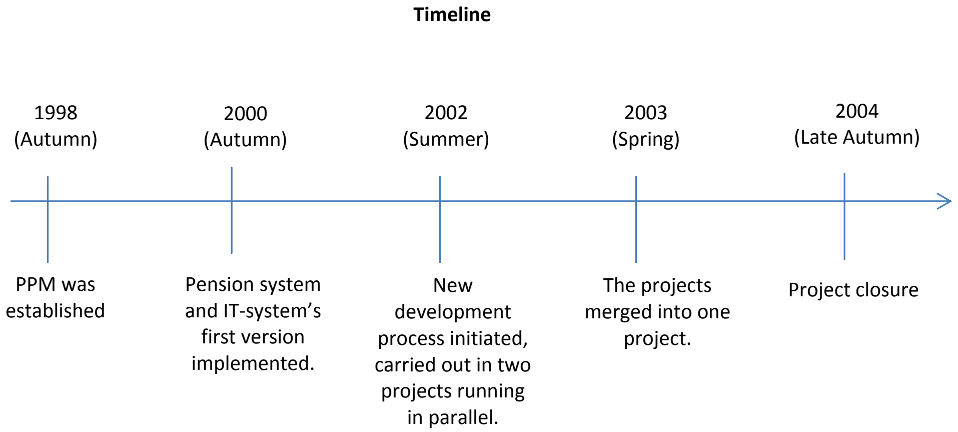

Timeline
1998 (Autumn)
2000 (Autumn)
2002 (Summer)
2003 (Spring)
2004 (Late Autumn)
PPM was established
Pension system and IT-system's first version implemented.
New development process initiated, carried out in two projects running in parallel.
The projects merged into one project.
Project closure

Figure 4: Timeline

1.2 Exploring and mapping a virgin territory

Much of the work at PPM during the first years concerned the elaboration of different pension and fund services. Pension savers should not only be able to choose what funds to invest in but also to retire and get the pension money. Furthermore, pension savers must be able to change funds how many times they wanted and whenever they wanted, as mentioned above. Moreover, something called "survivor's pension" should be offered to pension savers. In a way, services like these were created and mapped as different *processes*. That is, the "application for premium pension" represented one process and "application for fund choice" another, and "application for fund change" a third and "application for survivor's pension" a fourth one. Each process contained different steps to be carried out in order to complete the service. There were lots of processes to be created. In the figure below a short and simplified version of the "application for premium pension" process is shown.

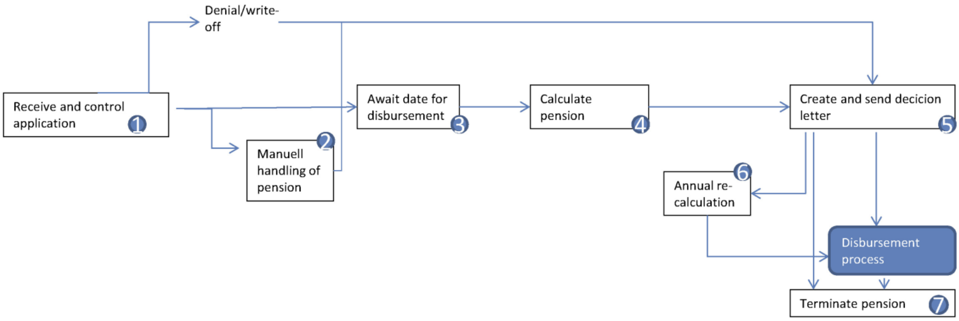

Denial/write-off
Receive and control application
1
Manuell handling of pension
2
Await date for disbursement
3
Calculate pension
4
Create and send decicion letter
5
Annual re-calculation
6
Disbursement process
Terminate pension
7

Figure 5: Application for Premium Pension

In this figure the process starts when PPM receives an application (1). Here in the first step the information of the application should be controlled. For example, the social security number should be controlled and verified, the signature, age of the applicant, address, the barcode number. If the application is incorrect or invalid it goes directly to number (5) where a decision letter is formulated (5) and the case is closed (7). Otherwise, the application is handled automatically or manually. If handled manually (2) different measures or *routines* should be carried out by a pension administrator. For instance, fetch the application from the application queue and open the case, check the application, decide what to do: for instance, complement the application, contact the pension saver, make a pension decision, send it to the queue for quality assurance; these are examples of different manual routines. Then, in step (3) the day when the money should be sent to the pension saver is awaited and in step (4) when the time has arrived the pension disbursement should be calculated, taking the value of the fund investment and the age of the applicant and expected lifetime into consideration. The next step, (5) is to create and send a decision letter to the pension saver. If this is done manually, the routine here says that the case and the decision must be verified according to the quality assurance principles that also were elaborated at the same time as the processes and the routines were created. In step (6) the annual re-calculation of the pension disbursement is done taking the fund growth, fees and age and lifetime calculation into account. This procedure continues until the pension saver dies. The money is disbursed in the blue step. The blue step represents another process, called the "Disbursement Process", which is not described here. Often different processes meet and are interdependent which make the whole system quite complex. When the pension saver has died the pension case is terminated (7).

As was stated above, this process description is a simplified version of the real process. Several steps and routines have been cut out, for example, the step in which PPM informs the pension saver about the premium pension disbursement, the step where the pension saver orders an application form, the stage showing the process way if a change application is sent in to PPM where the pension saver wants to change the monthly disbursement amount or change something else, and the step that shows what the process would look like if there is a survivor's pension connected with the pension saver, and the step regarding potential need of immediate recalculation of the monthly disbursement. Different routines such as when and how to order information from other subsystems, print reports, check financial records, put it on a pension administrator's "to do list", register changes in another system have also been left out in order to simplify the figure and explanation. The purpose here is to describe what a process and routine are and what it could look like at a basic level, since these terms were central in the work and are recurring throughout the case description. Still it is important to know that the processes and routines constituted a complex net of relationships and interdependencies. Identifying the processes and formulating the routines, both manual and automatic, was in way to create "the world of the premium pension" and constituted one of the main tasks at PPM.

The pension processes were elaborated with starting point in the pension law. Yet, since the law did not contain specific details, the jurists that worked at PPM must elaborate the law itself, the pension rights and the regulatory system at the same time as the agency's processes were outlined. For example, details of the law on how the pension money could be disbursed, pension rights, survivors' rights, from which day a pension saver could start receiving the money, how long time before retirement an application letter must have been sent to the agency, the conditions that the pension saver must fulfil to acquire the pension or survivor's pension must be elaborated and decided on in order to create the operational procedures of the pension processes.

The jurists and the pension experts at PPM must also investigate and decide what personal information details can be processed online, what the PPM worker and the IT-system should do if the application contains incorrect details such as if the pension saver's address is unclear or does not exist as stated in the application form, or what should happen if the pension saver has lived or lives abroad, does not speak Swedish or has applied for pension disbursements before retirement age, or if the pension saver dies and has no Survivor's pension insurance or no relatives,.

The pension law and the regulations and responsibilities of PPM should be precisely elaborated and described for every such circumstance. In addition to this, the pension disbursements and fund calculations must be developed and formulated, taking age, income, expected lifetime, growth figures and more into consideration. One big part of the work was thus to imagine what situations could occur and create a rule and routine out of that. The amount of possible occurrences made the task of extending the law complicated. Writing and clarifying the premium pension law and PPM rules and regulations constituted also a main task at PPM.

Taken together, this task was in a way very much to explore an unknown terrain and map a virgin territory of which knowledge did not exist. In addition, at the same time as the organizational members of PPM, that is, the jurists, fund, pension and communication experts, identified the processes and routines and elaborated the law they implicitly also called for IT technology improvements that at this time went far beyond the current state of technical knowledge at PPM. This will be explained in the next section.

1.3 Calling for technology advancement

More functionality in the information system was needed to administrate pension applications, fund choice and fund change requests with its different potential occurrences. In addition, the computer system must be able to handle and display plenty of information. For example, when managing a fund change request current fund choices must be shown, allocation proportions, transactions, development rates, cases in progress, and case status. Likewise, when processing a pension application, the application form must be shown on the computer screen as well as the pension saver's personal details, case type, id number, dates, support staff's notes of all cases. There were well over 5 million of pension savers accounts. In addition, the system must also be able to perform complicated pension calculations. Over 5 million of individual accounts with personal fund investments, unlimited possibilities to make changes in fund investments and traceability of all changes in each account implied that the system must be able to contain and process masses of information.

The 24-hours agency ideal implied further that the service level should be raised through improved web solutions and electronic-based communication over the Internet. The pension saver should be able to get information and apply for different services online and communicate with the agency 24/7. The accessibility to the agency should be developed through the possibility to contact the agency in different ways, not only through the traditional mail-way but also through different channels such as interactive website with personal log-in possibilities, automatic telephone service, and personal service over the phone. Furthermore, one part of the modernization of services was to make the pension and fund processes more automatic. The different pension applications and requests should no longer be handled manually, at least not to the same extent as before. The default scenario should be to process information automatically in the system; only special cases, processes or services and problems should be handled manually. In addition to these requirements there were also performance, security and stability requirements. For example, a fund change application should take no more than three week days no matter how many fund changes processed at the same time. The system must be built so that it operates quickly and accurately, never "goes down", leaks information or by accident let pension savers or PPM personnel do things that go beyond their rights.

The new functionality requirements, information requirements, traceability requirements, enhanced accessibility and personal service requirements, automatic processing requirements and performance and security requirements signified that the different subsystems (fund trade system, communication channels systems, information systems, databases) must be able to send and share information and communicate effectively with each other. This means that one of the most important technical improvements concerned integration of nearly ten different systems. Achieving this was a great challenge for PPM since it required knowledge that nobody had at the outset of the process - knowledge that was fast developing and changing in the IT-industry at this point in time. In sum, both the development of the premium pension law and its processes and the development of the information system involved uncertainty and exploration of virgin territories.

1.4 Commencing the journey

PPM consisted of six departments, the Insurance, Communication, Fund, Administration and Planning, HRM and the IT department when the development process started in the middle of 2002. Members from almost all departments were engaged in the development of the new services of the premium pension and the development of the new information system. Those members that were engaged in the elaboration of the pension law and “business” services and processes collaborated in a project that was called ”The Pension-Project” or “Requirement Project”. This project mostly consisted of people from the Insurance and Communications departments. Those who were involved in the development of the IT-system came from the IT-department and worked together in another project which they called the “IT-project”. (In the end of the thesis there is an appendix with the different people that will be quoted in the story below.) The intention was basically that “The Pension Project”/”The Requirement Project” should identify the pension processes and the routines and rules that must be carried out to perform the processes and then write down the requirements that they had on the computer system. The requirements should be handed over to the “IT-project” who in turn should build the system in accordance with these requirements. This may sound straightforward but as we will see in the next section the journey was barely commenced when different problems started popping up.

2. Problems banking up

”It was chaos the first year” as one of the developers said. During the first year so many troubles turned up so that the project members experienced that they did not achieve anything. The main problems that occurred from the middle of 2002 until May 2003 are presented one by one in this section. It begins with the problem with constantly changing requirements and disintegrated development of system solutions (2.1). It continues with the problem of living in different “worlds of meaning” (2.2), which is followed by a description of an attempt to rescue and improve the situation (2.3). This attempt made it yet worse and caused a problem concerning the power of balance between the project groups, explained in (2.4). In the last part of the section, (2.5), it is explained how an expansion of both the project organization and the scope of the task resulted in that the situation went out of control.

2.1 Moving targets and blind missiles

The work with elaborating and developing the pension processes was in a way to construct a “map” that for each day contained more and more places, roads, relationships and details. The PPM workers experimented and learned along the way about how to organize and administrate pension processes and fund administration, and what system support was needed and how to develop adequate system solutions. Early versions of processes, systems requirements and solutions became quickly obsolete and must be updated constantly. Requirements and program code was changed over and over again; first a few requirements were written and developed, then these were updated and changed at the same time as new requirements were specified, then the first requirements were amended once more and the program code updated and new systems versions developed, and so on. The iteration procedure was a natural and expected element of the process but at the same time it caused disorder, instability, uncertainty and non-progressive development circles, according to the developers. It was a real “circus” as one project manager said. The project members were walking in circles and the work did not progress. One of the developers commented that “everything was so fluid”, which meant that as soon as one step towards the goal had been taken, the “target” had changed with the result that the developers were pushed back to square one again. Several developers argued that ever-changing requirements were a big problem since it hindered them from progressing and completing the product. One of the Programming Leaders said:

> "There were no complete versions of requirement specifications. They changed all the time. We spent plenty of time updating things that were already designed and built because we constantly got amendments. It destroyed everything. We could not move forward."
>
> (Ben, IT Programming Leader)

Constant changes in turn had also a bad impact on the quality of the system in terms of instability and insecurity. In addition, when changes were made in the program code several parts must be retested, which easily could be very time-consuming. Moreover, interdependencies between different system functionality resulted often in that if a change should be made in one place, lots of other things must be changed too. Many of the requirements were formulated without cost and developing time awareness, according to the developers. One of the Programming Leaders said:

> "Particularly dangerous was that the whole fundamental framework that we built was put at risk because we repeatedly received changes in requirements that had an impact on the framework. And if you make changes in the framework you need to change everything that you have built based upon that. Then you must change everything. A minor change in one place that overturns the framework results in much much more work than what they calculate. We were not involved in specifying the requirements and did not have the chance to tell how changes influence stability, developing time, or what would have to be retested due to the change. So they did not know the costs. They did not get any feedback. There was no dialogue. There were some documents but they could be changed at any time without cost discussion. Changes in requirements were apparently always permitted and we simply had no choice but to deliver."
>
> (Ben, IT Programming Leader)

In addition to changing requirements and goals, that is, "moving targets", developers often took own initiatives and programmed technical solutions without underlying requirements; with no "target" in sight at all. It often happened that planned and developed solutions were changed when a developer came up with a better idea. Such "blind missiles" were thus not based on any specific "order" or "customer" request and did not support any particular business process. When developers worked independently in this way many intelligent solutions were developed but when it was time to integrate the solutions into one coherent whole it often turned up that pieces did not go well together. As with the case with moving targets, these blind missiles also resulted in that the process did not proceed; spontaneous changes of this kind mostly consumed lots of time and money. One developer said:

> ”The developers often programmed things without requirements. There were many changes in the detailed design that did not depend on changes in requirements, mere technical changes. Many changes constitute a big problem. It has happened several times that a decision on something has been made but then after a week of programming someone has created something better and so the

> process starts all over again. It might have been a better solution but in projects one must take time and money into consideration; it does not work to iterate innumerable times."
>
> (Adam, Developer)

Each developer worked on the part or detail of the system that he was interested in and no one took (or was assigned) the responsibility for the "whole". As one developer said "in the beginning there was much focus on 'finesse' in the solutions. Furthermore, the communication on who did what was sometimes absent or unclear and everyone could be involved in everything: one developer asserted that "everybody does everything". "Technicians sit in their corner and do not communicate before reaching the end of the project – then they start compromising and solving problems" one of the developers argued.

Due to the lack of overall responsibility, the overall construction drawing, or "design" in PPM and system development terminology, "emerged" as a result of the parts. One can say that it was like first making the pieces of a jigsaw puzzle one by one and then try to put the pieces together and hope that the pieces should form a picture. It was like start building a house before the construction drawings were finished. This resulted in a disorganized system solution hard to further extend and develop. The information in the system was illogically organized too. For example, there were different categories that basically held the same information and there was information that had been split in different categories but should better have been grouped together. In essence, this unstructured work and lack of in-group coordination meant that anyone could take the process in any direction at any point in time.

Finally, absence of coordination and overall system description resulted also in that they were unable to visualize a coherent model, simulation or prototype of the computer program to the members of the Pension or Requirement Project. This made it hard for them to understand what the computer program would look like as a whole before it was developed, which in turn resulted in that they could not confirm that the system would meet their expectations. The IT Project Manager reflected over the situation in hindsight and said that "IT:s ability to describe the solution was really poor. There was no organization for it." Requirement Project members did therefore not know what they would get before the solution was already, and sometimes in their view, wrongly programmed. As we will see in the next section, the lack of overall description of the IT-solution resulted in misunderstandings between the two groups.

2.2 Living in different "worlds"

The Requirement Project members often complained during the first year that they did not get what had been ordered. Requirements project members wanted the IT-project to give a response to their requirements in terms of a model, prototype, picture, design drawings or similar, to prove that they had understood what the Requirement Project members had ordered. There was no organization set up in the IT-project to formally and systematically receive and coordinate the requirements and "translate" them into overall integrated technical systems descriptions, that is, "system design" or "system architecture" that showed *how* the requirements would be solved. The fact that there were no prototypes or drawings or descriptions of how the developers intended to build the system and create the functionality that the Requirement Project members had ordered resulted in communication problems between the groups. Requirements project members did not understand what they would get, and became often disappointed and began to distrust developers' ability to deliver a well-functioning IT-system.

In addition, the Requirement Project members reasoned that the developers must show and confirm that they would solve the problem, not only because the deliveries had been inferior in the Requirement Project members' view so far, but also because of the fact that writing requirement specifications is difficult and that there always is a risk that different people interpret the requirements in different ways. The Requirement Project manager, who was hired as an expert consultant, said that PPM acted in general as if almost anyone was able to write system requirement specifications which she argued was wrong since not many members had knowledge, experiences and ability to see things from both a technical IT-perspective and a pension process perspective. The pension experts in the Requirement Project cannot translate their needs into IT-language and they cannot either judge whether the technical solution that the developers might suggest will match their needs and solve the problem. And the technicians do not understand the underlying pension processes and the problem that the IT-support system should solve, the Requirement Project members argued. The Requirement Project Manager said:

> "PPM acts like if anyone can write requirement specifications. There is a template that says how it is supposed to be written and then anyone should understand and be able to write it. I don't believe in this. It is a rather naïve way to look upon it. People are not very experienced in writing requirements specifications at PPM. There are not many people who know how to write requirements from this 'in-between perspective'. It easily becomes a very deep operations perspective, which is hard to translate. Those who write requirements do not know if the answer that they get corresponds to what they wanted because they do not understand the answer. And the one who receives the requirements shakes his head and says I don't understand what they want, this is not complete and this is not understandable. There is hence a too long road between IT and operations."
>
> (Elisabeth, Requirement Project Manager)

In the same vein, requirements could always be interpreted in different ways according to a Requirement Project member, since requirements do not constitute complete descriptions of the reality or an exact need of a real-life situation. First of all, the one who writes the requirement may have been thinking incorrectly or assumed certain things and missed out details in the pension process or filled in the requirement document in the wrong way. Then the developer may interpret the needs in a different way, assuming things that perhaps not have been stated or specified, and then when the tester should conclude and determine whether the solution matches the requirement and works properly he or she may understand it yet differently. The Requirement Project member said:

> "There are no exact requirements and hence there is always room for different interpretations. This means that in all instances there are potential pitfalls; in the first place, when a requirement is being created, the one who specifies it may think wrongly, interpret different process steps incorrectly and write it poorly. Then the developer may understand it in another way than the one who

> wrote the requirement had intended. Then the tester may interpret the requirement in a third way, so there are many latent sources of misunderstandings and troubles that we need to decide on how to deal with."
>
> (Kristina, Requirement Project Member)

From the developers' point of view there was also another dimension of the problem with misunderstanding. They argued that "Requirements Project" members must coordinate and prioritize the requirements in order to avoid delivery confusion and conflicts. The deliveries must be specified in terms of both content and time, that is, on which day each requirement should be finished. The developers asserted that they received the requirements too late to be finished on the day the Requirement Project expected. One of the Programming Leaders stated:

> "They [the Requirement Project] received a delivery but complained that certain requirements solutions were not included and we said 'that's right, but it depends on the fact that we got the requirements too late, there was no time to develop it'. And there was no priority among the requirements. There were some basic formal deliveries but not a description of what version of the requirements that should be delivered at different dates. They did not specify when the requirements were supposed to be delivered."
>
> (Ben, IT Programming Leader)

Hence, neither the IT-project members nor the Requirements Project members knew *what* or *when* to expect, which caused uncertainty and disorder. Developers and requirements project members lived in different "worlds" for almost a year. One project member declared:

> "What the IT-project has done is very unclear to the Requirement Project members. And the IT-project members must have been thinking that this is like a "black hole" or a "nothing"; 'what is this?' They do not probably understand anything 'what do you mean with processes, what routines, what???' That kind of feeling is what I have had"
>
> (Kristina, Requirement Project Member)

2.3 Rescue attempt

In order to reduce the uncertainty that lack of design sheets and prototypes induced and to minimize interpretation differences in the content of the requirements, Requirement Project members began specifying the requirements with more details. The intention was that more detailed requirements should result in that the IT-project delivered exactly what the requirements project members had ordered. It should also reduce the risk that technicians freely developed what they thought was appropriate and necessary. Some requirements were specified meticulously, which implied that "Requirements Project" members not only described *what* they wanted but also *how* it should be done, how the system should work and what it should look like. For example, in the K2-client (the system program that accessed and presented pension and fund information that was processed in a larger server system, explained more later on in the chapter) almost everything that the user would see on the screen was specified by the Requirement Project members. Nevertheless, developers argued that the neat exactness of requirement specifications involved a certain risk since those who wrote the requirements were not technically skilled and could not foresee technical consequences in terms of security, stability, developing time, cause-and-effect relationships, and future maintenance necessities with different requirements details. The IT-project Manager said:

> "The Requirement Project Members considered it hard to understand what they should get, which resulted in more detailed requirements. The requirements described very much how the system should work. As regarded the K2-system they described exactly what the client should look like and where exactly the keys should be placed and everything. There were very detailed requirements. This reduced the possibilities for developers to create cost effective and robust qualitative solutions."
>
> (John, IT-project Manager)

One designer asserted that if the developers build the system according to such requirement specifications the "customers" would often be disappointed since they would get things that do not work very well and that would also have a great crash-threat inbuilt in the system. In addition, it is wrong way of working; one should set and finish the big frame and the large pieces first and then move on with the details of the requirements, the designer argued. Compared with building a house, the ground and the main walls, stairs, roof, rooms, and the strength of the construction must be set before the interior decoration is made. The designer explained it in the following way:

> "The problem with too detailed requirements is that the clients get exactly what they require, and the problem with that is that they do not understand what they require. They do not understand the domain of technical solutions. And therefore they should not engage in specifying details. It is like they are saying 'we want this car with a grill in the backseat', which is of course a very good idea if you get hungry while driving, but you miss the fact that the car will burn down if we build it like this. In addition, engaging in details before the overall idea is confirmed involves the risk that something really crucial is forgotten. It is like ordering a car and specifying everything like colour, dashboard, different keys and buttons, top speed and everything but then forgetting the wheel or something as essential as that."
>
> (Simon, Designer)

Instead of elaborating the details during several months the developers suggested that Requirement Project members and developers should work more together earlier. The requirement specifications should only indicate what the system must be able to do at an overall level and what the users should be able to see and do and what functions that the system should have at an overall level. Then already in this tentative stage the developers should be involved to give their perspective on what might be feasible. The IT-project

Manager also said that an early dialogue where requirements and suggestions go back and forth between the Requirement Project members and developers would end in “solutions that cost less from a developing perspective but that will still work and be good enough for the users”. The developers would not need to program or implement real underlying program code but instead make drawings or use simple prototypes and explain in words how they intended to solve the problem. The developers said that their suggestions might even exceed expectations and the developers may well be able to deliver something that could be more user-friendly and superior to what the Requirement Project members could imagine. One designer assumed:

> “Requirements are more like ’this must work; we must be able to do this and get more or less this functionality’. Already here one should start communicating with those who should build the system and try to collect opinions and perspectives from them on what is good, instead of start guessing. (…) One should not specify for three months but have dense collaboration, communicating by result, and make simple drawings on the whiteboard. Use prototypes, for example the shell of the K2-system. We do not need to implement any underlying functions, just explain ‘this is what we intend to do’. It might be better than what the ‘customer’ had imagined and easier to use.”
>
> (Simon, Designer)

However, the Requirement Project Manager did not agree with the developers’ reasoning and said that they did not specify details for the sake of specifying details. In her point of view, each detail had an underlying meaning; there was is a point behind every requirement specification. Furthermore, several requirements members argued that the requirements only specified what the pension law and PPM:s rules ordered and were therefore not too detailed. A Requirement Project Member argued:

> ”It is the regulations that are described; ’if this happens, this and that must happen’ and that cannot be changed because it is the law that dictates what is written in the regulations and the rules of PPM.”
>
> (Kristina, Requirement Project Member)

Nevertheless, developers asserted that creating detailed requirements on this level meant that requirement people were involved in the developers’ job but the developers were not involved in the requirement work. IT-project members received requirements that they were supposed to build without foregoing discussion on feasibility and suitability. This continued during fall 2002 and springtime 2003 and turned into a battle of power concerning each group’s influence on the work process and outcome. This is described in the next section.

2.4 Imbalance of power

Developers argued that the Requirement Project members had too much power over the process and the product that they were jointly developing and that the imbalance of power resulted in suboptimal solutions that took more time and cost more money than needed. The developers’ competence was not taken care of and their contribution to enabling administration of pension processes based on high-tech solutions was neglected, according to developers and system designers. The Requirement Project members were the “owner of the process” and acted as if the IT-project members existed only to serve and help them accomplish their mission, that is, the agency’s overall political mission of offering modern pension services. The frustration among developers grew stronger and stronger during the first year.

> ”They [requirements project members] ruled over the drawings and sketches and there was nothing we could say. In the beginning when there was less time pressure it was like they simply said ‘don’t question, you don’t understand PPM:s pension and fund processes anyway so just build it’.(…) Operations has too much power.”
>
> (Simon, Designer)

One “Requirements Project” member who also was an engineer reflected over the situation and explained it in the following way. If “business” people or those who work on pension processes or administrative operations are weak, developers will start ruling the process and take over the wheel and develop things that are not necessarily needed. On the other hand if pension administration or operations people are in power much business needs and requirements will be created that will be very expensive to build and implement. The challenge is to find a good solution to handle this. The member said:

> “The risk is if the administrative side is weak the technicians will take over. And contrary, if the administrative people are powerful, plenty of requirements will be invented that will be expensive to realize, so one must find a solution to this. But I believe that the risk when administrative people are weak and technicians more powerful is greater. And then technicians can sell in almost anything and if one cannot question and specify exactly what one wants and why and how, it easily turns out to be a ‘technicians’ party’. It is known by experience that it can end up in this way since it is ‘so fun to develop new technology’”.
>
> (Jessica, Requirements Project Member)

Developers argued that a formal development method, rules and directives were needed to manage and balance the power between the two projects. The development process should be more regulated in order to accomplish reasonable requirements and systems solutions and avoid perpetual changes in requirements and solutions. There was a need for contracts, agreements and shared development plans. One Programming Leader said:

> “We should have rules, a formal way of working.”
>
> (Ben, IT Programming Leader)

The power battle continued at the same time as more system functionality was required and more program code developed. The informal way of working left much room for disputes, confusion and misunderstandings. In addition to the ambiguity in the relationship between the two project groups, both groups had also internal coordination and communication difficulties. Taken together, the interaction and coordination problems, the rising pile of requirement specifications and growing lines of system code, and as we will see, employment of more project workers eventually ended up in a crisis. The ship lost manoeuvrability. This is explained in the following section.

2.5 Ballooning a fuzzy organization

There were lack of formal interaction rules to direct the cross-functional work between the Requirement Project and the IT-Project. This caused power conflicts as described above and confusion in what should be done, when and how. When looking into the organization of the Requirement Project one could find some problems of coordination and planning within the group as well. The members explained that the structure and order in the way of working was not at all clear. Those who were creating the pension processes and the requirement specifications had much to do in a short time but no plans that showed when what should be done and finished. Many project members said that the work tempo was high and the time pressure strong. They often worked together in workshops but did not plan the meetings in advance or specify what should be done at what meeting. One of them said:

> "The tempo of the production of requirements was very high. The requirement document was sent over on one day and then we were supposed to read it at once and then review them the following day. Or often, we got the requirements during a meeting to review immediately."
>
> (Caroline, Requirement Project Member)

There were many people involved in the elaboration of the pension and fund processes and in the specification of requirements. One problem with that was that they all had opinions on the requirements which they individually discussed with the developers. Their internal coordination of tasks and responsibility and their internal communication on what had happened and how things proceeded did not work well. One of the Requirement Project members said:

> "'Oh oh oh, we must meet NOW and oops, oops, oops, this must be delivered NOW and we have not even reviewed it yet and opsy daisy, this must also be done NOW. We must get together, all of us now.' The work had this kind of character. (…) There should be a small expert group so that not too many people start running around spreading their opinions and deciding and discussing with developers 'I think we should do this instead' without reporting back to the one that should have been responsible for that requirement. If this is unclear, nobody knows what anyone does.(…) There must be a better team play; we cannot go in different directions"
>
> (Kristina, Requirement Project Member)

Nevertheless, the work continued despite the weaknesses in coordination and collaboration within the group and between the groups. As we know, the process started with identification of the pension application process and disbursements bus as time went by all types of processes and routines were created and system functionality specified and created to support the administration of the processes and routines. As they identified different processes and IT-needs they also reasoned that other things were needed and should be created at the same time. One Requirement Project member described it:

> ”We had been working for a couple of months and then we reasoned that ’if we do this we could do this as well’ and so it continued to grow.”
>
> (Louise, Requirement Project Member)

As more pension and fund processes were identified more IT-support and system functionality was needed. As a consequence, more system requirements were specified and more system functionality programmed. The task swelled up. Project members realized that the process was growing much bigger than they had ever imagined at the project outset in 2002.

> ”I did not understand in the beginning how big this project would become, probably no one did, I was prepared on doing the pension application and fund change processes but then it swelled gradually and we added the diary and everything should be included in this project. (…) It became much bigger than I had imagined.”
>
> *(Elsie, Requirement Project Member, Pension Unit Manager)*

In the IT-project the lack of solution descriptions and design documents resulted in that the internal coordination and communication within the IT-project among the developers became problematic as mentioned earlier. System solution alternatives, interdependencies and coherence of different solutions were not evaluated “on paper” or in “theory” before programming and when time came when the parts should be combined they did not play well together. Beside the coordination difficulties that the lack of overall and detailed design implied, it also brought quality and maintenance problems since new functionality and changes were hard to implement, track and evaluate in the program. One could read in a design review document written in May 2003 that system solutions had often been directly coded from the requirements instead of being created out of the design and architecture documents and that this must be corrected to ensure system quality and serviceability. Moreover, also in this project the time plans and management were unclear. The computer system should be finished in June 2003 but as one person at the IT department said “There was a date set by PPM that the system should be implemented in June 2003 but there were no plans that showed this”.

The net of pension and fund processes and the IT-system grew and developed in an uncontrollable way with hundreds of requirements and thousands of lines of code. According to project and programming leaders, PPM and the project board strived for top pension and fund performance and wide-ranging and complete system functionality but did not realize what different requirements involved in terms of technology demands, time and cost. Project members and programming leaders argued that one of the main problems was the board’s lack of priority; time, cost and functionality were of equal importance. The project managers argued that the board must give up something and asserted that as long as no priority is made among these steering parameters the process will continue forever, but it will not result in improved systems performance; in fact there was no progress, quite the opposite. The Requirement Project Manager said:

> "It is a little bit ironic; normally these three parameters are used to identify what can be sacrificed, but at PPM, at least in this project, nothing could be sacrificed. It was important to finish on time, it was important keeping costs down and it was important to maintain good quality. Everything was most important, which is an irresolvable equation. (…) It has been difficult to communicate and get understanding for the actual scope of the project and what they in fact wanted to be achieved. Time should be compressed to almost nothing but they still wanted to accomplish everything. There is lack of understanding how things work."
>
> (Elisabeth, Requirement Project Manager)

In order to manage the growing task PPM decided to employ more people. By May 2003 approximately 80-90 people were involved in the development process. However, the project managers argued that adding more project members does not work since it makes it even harder to collaborate, coordinate and communicate. The IT-Project Manager said that more people were hired but "I do not believe in that, usually it does not result in what one hopes for anyway" and likewise, the Requirement Project Manager argued that it does not get better:

> "If there are too many people it will be hard to manage anyway, and the communication will be bad anyway."
>
> (Elisabeth, Requirement Project Manager)

The ambiguous situation caused confusion and frustration. The way of working was problematic for both developers and those who elaborated the pension processes and the system requirements. "It was chaos the first year" as one of the developers said. The ballooning effect that writing more requirements and adding more system functionality and more people had on this fuzzy situation resulted in that the situation went out of control. As one of the project leaders said: "Before May 2003 nobody had understood the scope of the task". The development process was just about to fail when the project managers and the board eventually decided to completely reorganize the two projects and change work method in May 2003. This will be described in the following section.

3. Stairways to heaven?

The projects had been running for almost a year when it was decided to start all over again and change technology, organization, roles and way of working. These changes should hopefully lead to improved collaboration and communication and a more advanced IT-solution. The changes are described in the following sections, one by one. Practically everything became new as one of the project members stated:

> "The technical platform was changed and that was a really big thing. At the same time, we should create a new case management system and use a new work method and everything became new."
>
> (Elsie, Requirement Project Member, Pension Unit Manager)

3.1 New technology

After almost a year, in May 2003, it turned out that the pension and fund processes with the system requirements were bigger and demanded more advanced IT-technology than both project workers and PPM top management had understood. PPM came to the conclusion, by a technical review, that existing IT-systems must be totally reconstructed to enable and ensure required functionality, traceability, usability, security and serviceability. It was a critical decision since it involved a big change in technology; a new technical platform consisting of an integration point that should consolidate existing subsystems and administrate all information flows. A project manager explained that the development of the system functionality was just the "peak of an iceberg". The giant part of the task was beneath the "surface" and consisted of the development of the underlying machine that should integrate information from nearly 10 different sub-systems and administrate the workflows. This "engine" would be based on new technology that had not been used at PPM before. In addition, if the pension administrators should be able to quickly view and locate a case and also make amendments and see the status and history of a specific case, another more modern sort of user interface was needed. This is described and pictured below.

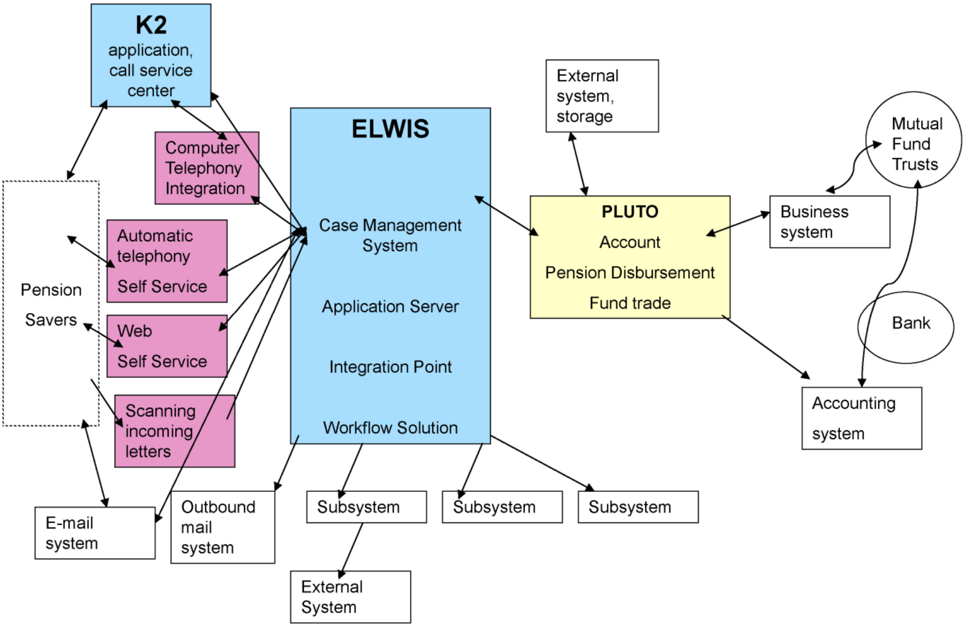


Figure 6: System Architecture

The figure shows the overall architecture of the IT-systems at PPM. The new technical platform consisted of the ELWIS and K2 systems, represented in the figure by the two blue boxes. The project comprised the construction and development of these systems. I will explain the most central pieces of the system architecture and how the system works from a pension saver and PPM administrators' point of view.

Let us say that a pension saver wants to contact the agency. He or she can choose among several different ways to do this. One way is to make a phone call and get in touch with an administrator at the call center (the small blue box). The administrator administrates the pension saver's request in the computer system called K2. There is also a "Computer Telephony Integration (CTI)" system connected to K2, a white small box below the K2-box in the figure. The CTI system means that the administrators answer the phone through the computer and can choose between various call operations through the computer and can also see the queue time and how many people are in line and other statistical measures.

If the person who made the phone call already has an account, the account details are shown in the K2 system. The K2 system gets the information from another system, ELWIS, the big blue box in the diagram. When the pension administrator at the call center types in information or data in the K2-system, it will be sent to ELWIS. K2 sends, picks up and uses information that is processed and stored in ELWIS. That is, the request that the pension saver has will be handled by the ELWIS system, for example the pension saver might want to get a brochure or a pension or fund application form, ask questions about the pension. ELWIS constituted a "case management system".

The pension saver can also contact the agency by sending a letter to the agency to be registered in the scanning system or by sending an email. He or she can also use the automatic telephony self-service system and self-service on the web. These different contact options that the pension saver has are represented by the pink boxes in the diagram and are called "communication channels" or "applications" at PPM. All the different systems send information to the ELWIS system. ELWIS integrates information and carries out different operations that the other systems "ask for". The communication channels send, pick up information from the ELWIS system and ask it to do different things. In this way, ELWIS serves the other systems. It is an "application server".

Moreover, since the ELWIS system integrates information from the different channels, that is, if the pension saver first calls the service center and then sends a letter as well as an email to the agency, all this will be put together by the ELWIS system in the pension saver's account. The administrators will be able to see all the pension saver's different requests and cases in one system, the K2 system, instead of having to work in and use one system for each channel and manually consolidate the information. Therefore, ELWIS is said to constitute an "integration point".

In addition, since ELWIS actually describes and sends information and is programmed to perform different pension and fund requests that involve various steps, it is also said to constitute a "workflow solution". In this work of processing information and performing different tasks the ELWIS system uses several other systems. Most critical here is the communication with the fund and account system, called PLUTO, the yellow box in the figure. In PLUTO the funds are traded and settled and registered on the pension saver's account. The account details can be changed here and the information given by the different fund traders are sent to PLUTO. In this system the pension calculations are carried out too (pension rights, disbursements). All information that exists in PLUTO is sent and integrated with the information in ELWIS and in turn, all information that the ELWIS system carries is presented to the user (pension administrator) in the K2 system. K2 is the window and Graphical User Interface of ELWIS. Here the information is translated and presented in graphical icons as regular computer users or "end users" are normally acquainted with, instead of lines of text and code language that the ELWIS system uses to communicate with the other systems and perform the workflows. Moreover, there are several connections to other subsystems and to external systems from both the ELWIS and Pluto systems, the white boxes in the figure, but there is no need to explain all these relationships and system connections - it is enough to see a picture of the complexity of the IT-architecture and know that the ELWIS system should be developed to communicate and integrate information from approximately 10 different systems. This is basically how the IT-architecture should be built and how it should work. The project task was as mentioned to create the K2 and the ELWIS system and ELWIS' interface with the other systems.

The change in IT-technology also implied that the pension processes and the requirements of the system also must be reconsidered and updated to make sure that the processes, the requirements and the system's information flows and functionality match and conform. Furthermore, the project board and the project managers argued that if PPM should continue the work considerable changes had to be made to solve the organizational problems that existed in and between the projects, so in May 2003 the way of working and the two projects were totally reorganized. The project members said that they made a complete "restart". This is explained in the following sections. First, the new organization is introduced, and then are the new roles that were formulated described, followed by an explanation of the new work method. Finally, the section ends with a description of the "new deal". Together, the changes seemed promising, according to the project workers.

3.2 New organization

To solve the problems with collaboration, coordination and misunderstandings between the projects and stop the discussions on whose fault it was when system functionality did not correspond to what the "Pension or Requirement Project" members desired, it was decided that developers and requirement people should be organized and placed in the same project and have one main responsible project manager. They should row the same boat and report to the same captain. Current project managers got new expert roles, explained under the next subhead, and a new person, a consultant, was hired as "Head Project Manager". The "we-against-them" atmosphere in the projects was expected to fade away by sharing responsibility and having a common project goal. The dialogue between the groups was expected to be stimulated and improved in such a setting. The previous Requirement Project Manager said about the new organization:

> ”It gives a 'we-feeling'. One will not deliver something into a 'black hole' and receive something else from another 'black hole' that not at all is what one expected. Instead, we are a group of competences rallied around a problem that must be solved. I think it is a good way of working. It involves less prestige since everyone has the same goal. People cannot engage in mudsling because it will not take anyone anywhere."
>
> (Elisabeth, former Requirement Project Manager)

Beside developers, requirement people and a new Head Project Manager, the new project organization also consisted of a group of test people. Their task consisted of testing the system to make sure that it worked properly and had the functionality it should have on the implementation day. The members of this group came from the line organization, more specifically the "Test Unit", which constituted a part of the IT-department of PPM. An external Test Manager, a consultant, was employed instead of using the existing Test Unit Manager, to organize the test process. The Head Project Manager was keen on delivering good system quality and did not want to take any risk that system bugs or non-working systems solutions were implemented, or that users would receive a system that they did not understand or were satisfied with. The Head Project Manager said that he wanted "one person to be responsible for the whole test process" instead of having different managers responsible for different subtests and subsystems, as they usually had in the line organization.

Moreover, there were some employees appointed to plan and prepare the PPM organization for the launch and implementation of the new system. This group's tasks were not clearly specified or thought out in detail and their tasks turned out to coincide to a large extent with the tasks of other groups. Mostly however, this group dealt with data conversion, which entailed moving information from the old systems to the new one. The new organization is summarized in the figure below.

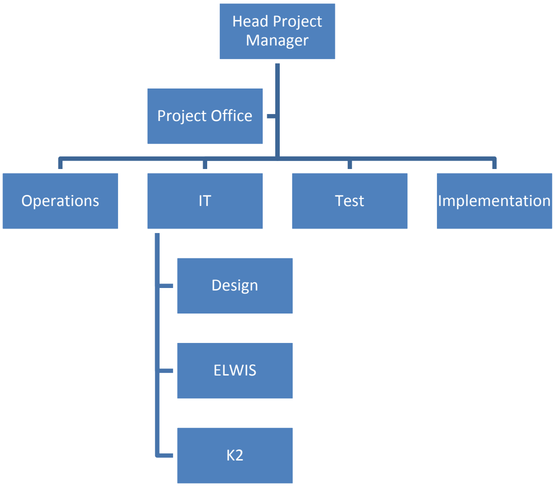


Figure 7: Project Organization Chart

At the top is the "Head Project Manager". Below him are four different groups, or "Subprojects", as PPM called it; IT, Test, Implementation (those who dealt with data conversion and system implementation planning) and Operations. Operations is the previous "Pension and Requirement Project". Each subproject was led by a Subproject Manager. The IT Subproject was in turn divided into three groups; a "Design" group and two different programming groups, "ELWIS" and "K2". Each group was led by a Programming Leader. The Design group constituted an important change in the organization. This group should solve the communication problems between those who ordered the IT-support and those who programmed the system solutions. The designers should create drawings and suggest solutions and present it to those who wrote the requirements to make sure that the developers programmed and delivered an IT-system that matched the needs and expectations of those who ordered it. This group was in PPM terminology accountable for the *overall or general design* of the systems and the interface between the systems. The other two IT-groups, K2 and ELWIS, programmed the K2 system and the ELWIS system respectively. The K2 system was the user interface, that is, the computer program or window that end users would see on the screen as described previously. As mentioned above, ELWIS constituted the server and database that processed and administrated and integrated information. The K2 system picked up (and delivered) and presented information that was processed in the ELWIS system.

Additionally, a project office consisting of two people was formed to monitor and proactively control different steps of the new development method (which will be described below). The project office should make sure that project members followed the new directives and rules. The office members administrated documentation, and watched over agreements and handovers between the different groups. This should make the process less ambiguous, increase quality and reduce changes and break the vicious "never-ending" iterations.

Also, for cost accounting purposes and in order to make the ship manoeuvrable again the number of project groups and members were reduced. For example, the developers of the communication channel systems (web, scanning, mail, telephony) were not included in the new project organization. Instead they should belong to the line organization again. Nearly 50% of the work force of the agency was involved in the two projects, approximately 90 people, during the first year. This number was reduced to about 35 after the restart of the project in May 2003.

Most project members were also placed with physical proximity in the same office area to facilitate coordination and strengthen collaboration willingness. Some project members argued that a long physical distance causes "we-and-them" feelings and makes it hard to understand what "the others" really are doing, why things take such a long time and it becomes easier to blame others. Several members said that low and bad insight in what other groups do was a general problem at PPM. People did only see their little task isolated from the rest. One of the project managers underscored the importance of being placed together especially when collaboration difficulties turn up. The previous Requirement Project Manager even assumed that being placed together is a pre-condition to make it work. For her it was important to hear what people talked about. Sitting together in the same office landscape makes it easier to come by to ask questions and chat or discuss something; both the physical distance and mental hindrance is reduced if people are placed in the same room and know each other's faces. Likewise, the design leader emphasized retrospectively the physical closeness to foster a team-spirit:

> ”It has meant very much for us to sit together. We did not do that from the beginning. It is extremely important in order to get the feeling that we belong to the same team.”
>
> (Simon, Designer)

Beside the new organizational structure, there were also some new roles created which turned out to be crucial for the project's progress. These specific roles and their tasks and responsibility are introduced below.

3.3 New roles

In the section above it was mentioned that instead of appointing one of the current project managers to become the Head Project Manager of the new project, an "unbiased" person from outside PPM was recruited as Head Project Manager. He placed himself among the developers, designers, testers and subproject managers in the open landscape of the IT-department. From this position he could hear the daily talk and follow the process on a daily basis. He had IT knowledge as well as experiences from pension and fund insurance processes. The previous IT-project Manager argued that it was important that the Head Project Manager had knowledge and experiences from both domains. Otherwise he would not be able to make the right decisions. In this situation things must be deselected and cut away to reduce the scope of the task. Without insights in both areas one cannot understand what and how to prioritize, or set up rules and directives, he assumed.

Moreover, in order to solve the problem with too many and continuously changing requirements that too many people took responsibility for and discussed with developers without internal coordination, a new role was created called "Requirement Coordinator". The Requirement Coordinator should be responsible for coordinating the requirements internally among those who were involved in the work of identifying pension and fund processes, routines and requirement specifications. There had been several different groups and people involved with many different interests and demands and opinions regarding the system functionality. The Requirement Coordinator should bind them all together and translate the needs and opinions into well-aligned system requirement documents. Then the Requirement Coordinator should discuss the requirements with the system designers to get their view on them before programming. In this way she should constitute the communication channel between the pension and fund administration people and the developers. The previous Pension and Requirement Project Manager was assigned this role and a new project leader was employed to the "Operations Subproject" (former Pension and Requirement Project). The Requirement Coordinator described her role in this way:

> ”The role has been to write requirement specifications based on what we discussed during the first year – we talked very much about what we wanted, and what it should look like and how it should work. So it was about to bind the operations together. That is what I consider most important. My role during the later stage of the project was to coordinate the requirements and manage the requirement specification process and make sure that the requirements were reasonable. Then discuss with the designers how the requirements were intended to be designed to make sure that the main part was done as we had imagined. That is, very much of a communication link in the project between IT and Operations."
>
> (Elisabeth, former Requirement Project Manager, Requirement Coordinator)

Instead of letting anyone from the "Pension and Requirement Project" discuss and formulate requirements with any developer, only the Requirement Coordinator should write and discuss requirements with the IT-project. The communication went from spontaneous and informal to planned and formal Even though the communication became highly restricted and canalised project members from both the IT and the Operations assumed that it was necessary to formalise the communication process in this way. The new Project Manager for Operations argued that this measure was absolutely needed since the requirement specifications had been so criticised. She said that this made the situation clearer and more stable.

At the IT-side a similar measure was taken by establishing a new group of IT-designers (mentioned in the previous section). Instead of allowing any developer discuss the requirements with anybody from the former "Pension and Requirement Project", only the design leader should communicate and discuss the requirements and potential system solutions with the Requirement Coordinator and create a complete solution suggestion, a design. This was an essential change in the project organization. Before this group, no complete system design had been made, which we remember can be likened to building a house without drawings. From the restart there were three people assigned to only create system design drawings. The previous IT-project Manager (who became a designer) stated that it was a huge and necessary job:

> ”We were three people who for 1.5 years only worked with the design descriptions on how we intended to build this and it was a big project so of course it was essential. We did not have these people from the beginning.”
>
> (John, former IT-project Manager, now Design Expert)

Another change in roles in the IT-project was the appointment of a new IT Project Manager. This person’s task in the beginning was to create detailed plans that showed how and when the work could be finished. After about 6 months this person quit and the Head Project Manager took over the role as IT Subproject Manager at the same time as he continued as Head Project Manager.

Furthermore, the new project office consisted of two people, of which one was engaged in the creation of overall project plans and administration. The other person was called “the police” by the project members, since he severely controlled that the new formal work method and development process were being followed. The design leader said that “he checked the documentation and acted a little bit as a police”. Project members admitted that this role was important even though they sometimes considered the controls and the formality of the process somewhat inconvenient. One of the project managers said:

> ”I believe that it is a success factor to have this kind of role because this [documentation, rules, formal procedures] is that kind of things that one easily neglects. He worked hard to keep some control over us. Herman has been involved too and kept control over the plans but he has not been as proactive in the same way as Oscar has been. Oscar has in a way chased us a little bit and forced us to do things while Herman is more like ‘this is the way it is’.”
>
> (Jessica, Requirement Project Member, Implementation Subproject Manager)

The new project organization and the new roles seemed to clear things up for the project workers and calm down the situation. The new work method and the new project directives constituted the most difficult change to implement on the other hand which made the project officers’ role very important in this project. The principles of the new work method and the new development process that the “police” was supposed to look after are described in the following section.

3.4 New work method

The most important change in the way of working was that the new Head Project Manager implemented a sequential waterfall-like development process model. The purpose was to stop the loops in requirements and system code and to make the different groups’ responsibility and the overall view of the project task clearer. It would furthermore be easier to manage the process when one thing at the time was carried out. The phases or steps in the "waterfall" process included at a general level "Requirement", "Design", Programming and Coding", and "Test", which would end in the final product and implementation of the system. The process flows downwards like a waterfall through the different steps as the figure below shows.

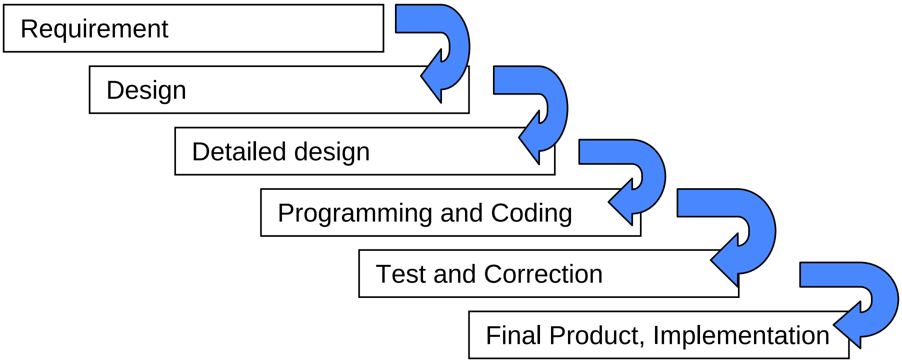


Figure 8: The waterfall model

One step should be finished before the next step starts. Nevertheless, in this project the steps overlapped to some extent because the one who would receive what the preceding group had been working up should have had a say on the work and be allowed to make some changes in order to, in turn, be able to do something valuable in his phase.

The first the Head Project Manager decided was to stop current programming since the requirements were still in progress. Then when the requirements had been written down and documented formally the requirement specifications should be reviewed so that the members of the "Operations" project could coordinate and agree internally before sending the documents over to the designers. This became a formal procedure, in this project the agreement was called "handshake". In the next step, instead of handing over the requirements directly to the developers, as they had done before, the requirements should be sent to the designers who should give their opinions on the requirements and also suggest a complete system solution. This solution should be written down in the so called "design documents". The Requirement Coordinator should review the solution suggestion. The requirements that specified *what* functionality the system should have, and the overall design, which described *how* the system would be built, should from the restart be formally documented, reviewed, accepted and signed by both "Operations Project" members and developers. The requirements should only describe the process flows and the routines descriptions should no longer serve as a base for the requirements. This signified that the requirements would not be as detailed as they used to be. When the Requirement Coordinator (and the members of the "Operations Project") and the designers agreed on both the requirements and the design, these documents should be "frozen". This was an important change in the process compared to before the restart and implied that no more changes in nor requirements or design documents could be done. After the signing, or "handshaking and freezing" as it was called, no uncontrolled changes were permitted. It meant that project members could not update, improve or change a thing individually and spontaneously. Instead, changes must go through a formal change procedure, in which developing time, stability, test time, alternative solutions, costs and time aspects were discussed. Then a formal change decision should be made based on that discussion. Previously, project members made changes more independently and informally as they considered appropriate. This change in the work model was welcomed by most project managers and leaders. The Requirement Coordinator said:

> ”I think freezing is good – these moments when we can say that we have at least come this far now.”
>
> (Elisabeth, Requirement Coordinator)

When the overall design of the system had been completed the work with the detailed design should commence. The overall design described how information travelled through the system and the detailed design specified different elements of each step in the system's processes. For example, when PPM receives a fund change request the application form should be scanned and then registered in the system. Then information is sent to a subsystem to actually perform the fund change. These are steps in the information flows, described in the overall design. In the first step, some details of the design could concern for example where the scanned application form should be visible on the screen, and where and how other information should be presented, that is, personal account information, fund holdings and fund information. The overall design became a document that constituted a general "frame" for all the details of the system solutions. Again, it can be compared to a construction drawing of a house. The detailed design would constitute the interior decoration of the rooms in the house. This "frame" meant that developers must build the system according to the general drawings and keep the details within the "borders" that the design sheets set. Thus, the design bounded and directed what developers could do in the next phase. The Design Leader said:

> ”The design becomes a steering document.”
>
> (Simon, Design Leader)

The developers should describe and document how they intended to build the details of the system before they started the actual programming. According to different IT-leaders, developers must explain and communicate how they planned to build the system in order to prove their understanding of the requirements and the users' needs. This documentation of the detailed design was a new step in the development process. Some developers were not very enthusiastic about this new step and preferred to just program and code the system solution directly from the requirements specifications or from the overall design documents and skip the detailed design description and documentation. But the previous IT- project Manager explained the importance of detailed design descriptions:

> ”If there is an agreement with the Operations Subproject Members on building something in a particular way, we have to implement it in that way too. The developers must understand it and make sure it coheres from the requirements all the way to the programming and coding, and the only way they can prove that they have understood is to describe the task and the solution. This was very problematic in the beginning; many of the developers were very unfamiliar with this. We wanted to know before they hit the computer keys and start writing the code that they had understood the task and what the "Operations" wanted.

> Otherwise it will be too late – when the code is written the train has passed.”
>
> (John, former IT-project Manager, now Design Expert)

Thus, in accordance with the waterfall model, no details of the design should be set or programmed before the overall design was finished. The detailed design was then expected to be completed and confirmed by the designers before the coding and programming started. After documenting and agreeing on the detailed design, coding and programming should start. The previous IT-project Manager said:

> ”You should never start a project like this before you know what to do. You should put effort and time in a requirements and analysis phase so that you know exactly what you want to accomplish. Then there should be an extensive phase when the IT-design and architecture is created. All those things must be set before programming.”
>
> (John, former IT-project Manager, now Design Expert)

Then the code and program components should be verified before formal delivery of functionality to the test group. The test group is supposed to start with application tests (entire program functionality) and continue with system integration tests (how different systems communicate and work together) and performance and stress tests. During this process system bugs and errors should be documented and formally reported to developers in a certain computer system to be corrected in a controlled manner. These tests and all corrections must be done and the system code should be finished, stopped and “frozen” before the final tests start, that is, acceptance test (clients evaluate and confirm that their requirements are fulfilled). When the system had been tested and accepted it is ready to be implemented in the organization.

The rules of the development process were described in the “project directives” called “the method handbook”. The Head Project Manager assumed that this work method usually demands lots of time and is a tardy way of working but when working with systems of this size and complexity one has no better alternative. And in this case the developers had begun building before the drawings were ready which resulted in that they had to start all over again and go back to the first step in the process. The Head Project Manager said:

> “It was one way to go. The requirement specifications were not ready and no one had ever done an overall design or a detailed design. It is a slow way of working, it takes time, but when working with systems of this dimension and complexity, this is the only work method we can use. We must work sequentially.”
>
> (Thomas, Head Project Manager)

The new method included as we have seen also more standardized and formalized documentation and communication. Requirement specification documents, design, detailed design sheets, deliveries, installation instructions, test cases and protocols, errors reports and changes should be written down and archived in accordance with certain templates, version policy, and standards. Furthermore, formal weekly meetings were set up to complement the informal communication and better organize progress controls, status updates, discussions, decision-making, and problem solving. The Head Project Manager had an individual meeting with each project manager every week on the same day at the same time. The project managers were supposed to hand in a written "status report" to the Head Project Manager before this meeting. During the meeting current status and problems were discussed. One of the project managers said about the status report and this kind of meeting:

> "It has been mandatory for each project manager to hand in a status report every week, which has been very good. It makes one shape up and think 'what have I done? What do I plan to do?' It makes one catch different balls that one might be about to lose. It is very good for one's own "shape-up"; one can feel the pressure. I cannot write week after week that I soon will start with a certain task. It is a good way of working. It gives the Head Project Manager an overall view of each sub-project and directs focus towards problems and difficulties that one might have and it makes one shape up. I believe it is a very useful model."
>
> (Jessica, Operations Subproject Member, Subproject Manager Implementation and Data Conversion)

The Head Project Manager began to have weekly meetings with project managers and project office members together as well. The project office members wrote meeting minutes, checked the development plans and controlled that decisions and work procedures followed project directives. The project managers argued that these meetings were a good way to get an overall understanding of the project and the different groups' internal progress and problems. The project managers' could furthermore relate their problems to each other and detect interdependencies in resources and planning of activities. Furthermore, project managers' meeting was always directly followed by a meeting regarding problems in requirements and system solutions that needed correction. Here at these meetings, problem analysis discussions and decisions were made in collaboration with invited experts.

This change at the overall project level towards more formal communication also resulted in that the communication became more planned and structured in the different subprojects too. The different project managers started to plan and set up time and dates for meetings with their group members. Among the "Operations" project members this change was welcomed while the implementation of status meetings in the IT-project was more problematic. In the Operations project the planning and meetings schedule were made by the new Operations Subproject Manager. The meetings were utilized to work something out together; these were more like workshops occasions than status updates meetings. The members of this subproject considered that the planning and scheduling was absolutely needed. The developers were more critical towards the planning and the status update meetings. In the IT project much time was spent on creating extensive development plans with specified milestones and deliveries that showed how and when the goal should be accomplished. Activities were specified one by one (content, time and developer). During the status update meetings the plans and the activities were checked but no problems were discussed or solved during these occasions.

Taken together, the project managers expected to gain control over the process again by the implementation of these changes, and so restore stability and break the never-ending treadmill. The previous IT-project Manager said:

> “We clarified things with clearer roles and a rigid structure, and clear handovers and deliveries and measures of progress.”
>
> (John, former IT-project Manager, now Design Expert)

3.5 New deal

One important aspect still remained to be considered though. The project board and PPM’s ambitions of accomplishing the “Grand Opus” and the fact that they had not been prepared to cut down on anything. They wanted a feature-rich IT-system that should serve the pension savers and the pension administrators at PPM in an excellent way and this should be done at the lowest cost possible as fast as possible. This was something that the project managers called an irresolvable “equation”. The new Head Project Manager was now expected to handle this. The resolution became the “new deal” and this is described in present section.

According to several project managers and members the Head Project Manager was not afraid of changing things in the situation that he did not consider work very well. He took a firm grip of the situation and made powerful decisions and changes in the way the work was run. It made a few people upset, doubting whether his authoritarian leadership style would be the most appropriate way ahead but mostly project members said that the changes were good and necessary. The new Head Project Managers also compelled others to choose and make decisions when things could not be coordinated or combined. For example, the project board had to prioritize and choose one steering parameter. The project had been running for such a long time, almost a year, and so far nothing useful had been delivered. In addition, PPM had experienced many troubles with the IT-development long before this process started so now members and managers from all different departments were worried that the project would fail and they would end up in a new costly affair; money and time and resources would then have been wasted again. The board members, who were also PPM’s department managers decided that time was most important, that is, to finish the project as fast as possible. One designer said:

> ”The date became holy. I think they wanted project closure more than something that would last for a long time. But that was good. Before Thomas there was the discussion that it had to be finished on time but we cannot lift anything out and it cannot cost more money. He said ‘drop two parameters, you can only keep one’. So that was good. What happened when Thomas came in was that he curbed people and made them take a stand and choose; upwards, downwards and sideways. His acting was consistent. Before him, there was a ‘kick downwards system’.”
>
> (Simon, Design Leader)

However, as a consequence of the decision on using time as steering parameter many hours and much effort was spent on creating time plans to see how and when things could be finished. Several project members and managers fought a lot to make the planning sheets point at the special day that the board wanted the project to be finished. The scope of the project had to be reduced which meant that the requirements had to be cut down. The previous “IT-project Manager” said:

> ”It felt like the planning almost killed us. We had this day in front of us and a bunch of things that must be done. We tried to squeeze, strain and puzzle with the numbers to make it on this desired date. It did not work. What one must do is to lift things out of the project. That was the whole turning point.”
>
> (John, former IT-project Manager, now Design Expert)

Moreover, the previous IT-project Manager asserted that there was also another reason why the scope of the project must be delimited - the shift in technology and the new technical platform. It constituted the largest part of the work and at PPM there were not many people with experiences from this kind of work and technology which implied that the developers must experiment and learn about the technology along the way. The former IT-project Manager underscored the importance of reducing the task and making it clear in such a situation:

> “We did not have enough competence either to run such a big shift in technology. We had a few people who had worked with this technology before but except for those there was not enough competence in the house, which implied that we had to build competence along the road. In such a case, it is important to delimit the scope of the project considerably and make the task clear before start working.”
>
> (John, former IT-project Manager, now Design Expert)

One of the greatest challenges became thus to limit the project task and set the frames for it. Beside the cutbacks in the requirements on system functionality, different technical aspects regarding performance excellence were also sorted out. The Design Leader emphasized that it was the date and not functionality or serviceability that was most important; “it was strong managing on the date”.

Nevertheless, the Operations Subproject Manager was not as convinced as the developers and designers and managers that cutting requirements was unquestionably the right way to go. She said it was “slimming for the sake of slimming” and that it was shown later on in the process that several things that were replaced with easier solutions became insufficient and that they in those cases had to add support-functionality, which also turned out to be insufficient and so even more support-solutions must be implemented and so on. It would have been better to keep many requirements as they were from the beginning, she believed. She said that people did not think through things very well. And at the same time the Operations felt the pressure that they had to cut more and more functionality. In addition, while some project members said that the strong focus on the date and keeping the time plan and not letting the plan “move” was necessary and good for the project, others said that having time plans with no slack or extra room for problems, creativity, vacation or sickness just made the plans unrealistic and absurd.

To sum up, people were in a way exposed to less uncertainty and pressure as a result of more structure, order and higher predictability in operations but in another sense, project members were also subject to more pressure in terms of compressed time frames and renewed collaboration and coordination expectations. It seemed though that the restart and its changes mostly brought a collective sense of relieve and ease; something people considered necessary and generally agreed upon. The Requirement Coordinator said:

> "By tradition, PPM have many people who are very interested in details of solutions rather that reflecting over if management and control works out well, or how we integrate different things so in a sense, I believe that the first part of the project would not have been so terrible if people had been interested in such things. The IT-work was very good after Thomas came in and took over. Before that, it was very messy, disorganized and complicated and unclear who did what and how it worked. It became clearer who had which role. Also the structure improved in this part [IT] of the process, which was the most difficult one to manage and organize, so it was a great difference."
>
> (Elisabeth, former Requirement Project Manager, now Requirement Coordinator)

The project managers and leaders were grateful for the fact that the new Head Project Manager protected the project workers from conflicts and discussions with the line organization. They experienced that a certain peace and silence emerged when the decisions and changes had been made and things stopped "moving". Project members reported that things became more stable and that they started to see how things could be coordinated and organized. The situation clarified. One project manager said:

> "I think everybody was extremely grateful for some structure and firmness, nobody questioned if it was good or bad. It was the contrast to how it was before which made people appreciate it. Just the fact that someone pointed out the goal, the way and when we should be finished - I believe it was exactly what people wanted; order. There was no discussion about it."
>
> (Jessica, Subproject Manager Implementation)

In the coming three parts we will see if all these changes worked as a "stairway to heaven", that is, if the project members managed to solve the task and end the project successfully. In the description of how the work unfolded special attention will be paid to problems and imaginative solutions in the way of working and in the way people integrated their different views and knowledge. The text is structured in accordance with the steps in the waterfall model. First in Leg 1, I will describe how the work with the requirements improved after the restart and how the project members managed to integrate the "Operations" members' knowledge and turn their "vision of peak performance" into stable and reliable pension and fund processes, routines and system requirement specifications. I will not go through every incident or report what happened from day-to-day, instead I have chosen parts of their work that were particularly interesting and constituted a typical pattern of interaction.

Then in the next part, in Leg 2, we will see how the designers and the developers managed to create the system drawings and turn them into a well-functioning system device. Last, in Leg 3, we shall see how the process unfolded when the testers should check the system at different levels and make sure it had been perfectly debugged to the launching date.

4. The first leg – from vision to specs

After the restart, the scope of the project was reduced and the goal became clearer. Changes were made in the organization and in the way of working – a strict sequential work model should now be applied. The first leg in this model was to turn "the vision" of what pension and fund services PPM should offer into coherent and fixed operation descriptions, procedures and IT-system specifications. This challenge is described in 4.1. There were many different people from different departments involved in this process which implied that various skills should be used and integrated, and differences in meaning and interests revealed and solved, which is described in 4.2. The project members used two different ways of working to deal with this challenge. These two ways are explained in 4.3 and 4.4. Last in the first leg in 4.5 is a description of the collective review procedure of the requirement specifications that should reduce and hinder future changes in the requirements. When this procedure has been completed the designers and developers pick up the baton and start the designing and programming work.

4.1 Aligning processes, routines and requirement specifications

The work with PPM operations or processes, practices and IT-support specifications had previously been rather disorganized and the pension and fund processes, the routines and the requirements were unclear and incomplete. One problem was that the overall view of the pension and fund processes and the system requirements was missing. The Operations Subproject members said that in the beginning of the project processes and routines and requirements were worked out, in that order. However, according to the Operations Subproject Manager, most time during spring 2003 was spent on elaborating the system requirements specifications separately and before the elaboration of the processes and routines had been finished. This meant that some requirements or IT-necessities were written without being sufficiently grounded in the organization's (PPM) operations and practices (processes and routines). It resulted in that the requirements were more elaborated and detailed than the processes that constituted the reason why the computer system was developed. As a consequence, the requirements and the routines and the processes did not longer match. To see the overall need for IT-support the project members must align operations, practices and IT-necessities. The Operations Subproject Manager argued that processes, routines and requirements needed to be elaborated at the same time, more in parallel to make them correspond to each other; one must go back and forth between these parts to ensure perfect correlation. The Operations Subproject Manager said:

> "One cannot finish processes and routines and then start with the requirements. Because when working with the requirements one works at a detailed level (…) and then it is important to go back and see if things still go together at the overall level and if it does not, one must investigate whether there was something wrong in the system requirements or if there was something wrong in the process drawing. One may then detect that 'this does not work; we forgot a very important law requirement which implies that we must take this route on the overall level and that involves these changes in routines and requirements'."
>
> *(Kristina, Operations Subproject Manager)*

This meant that the project members needed to create the whole picture of all processes at the same time as they worked out the details of the routines (how to carry out the processes) and identified what IT-support was needed for each step in the processes. During fall 2003 the "Operations Subproject" members started a comprehensive work where all processes, routines and requirements were created and updated and completed simultaneously. The figure below is a simplified example of how the processes and routines and system requirements are interrelated.

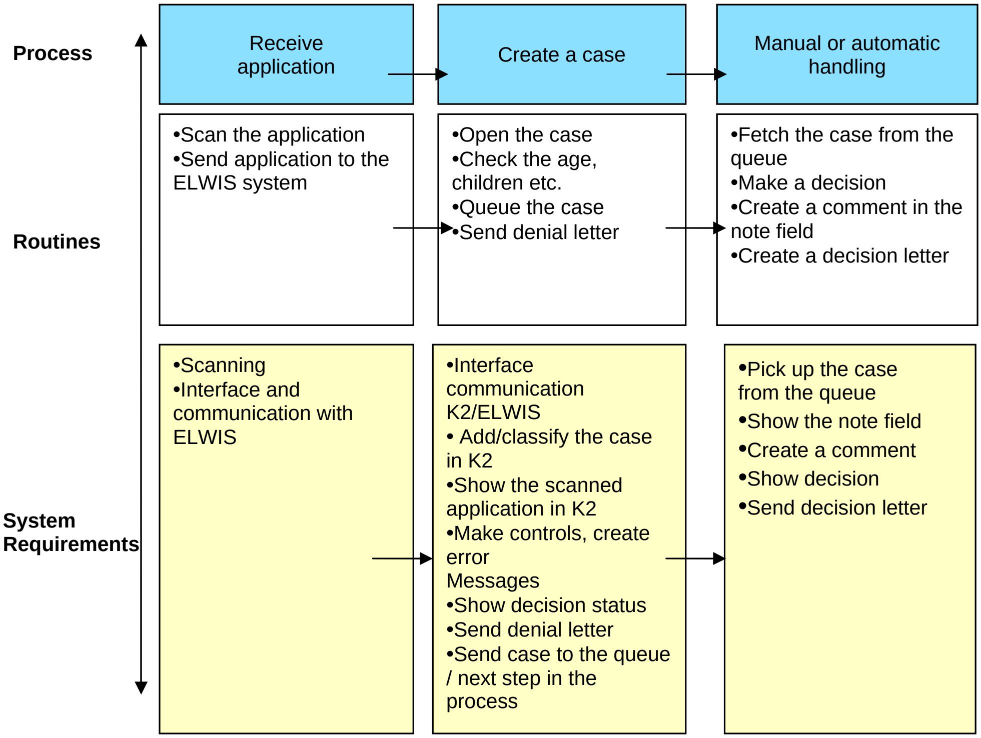

Process
Routines
System Requirements
Receive application
Create a case
Manual or automatic handling
•Scan the application
•Send application to the ELWIS system
•Open the case
•Check the age, children etc.
•Queue the case
•Send denial letter
•Fetch the case from the queue
•Make a decision
•Create a comment in the note field
•Create a decision letter
•Scanning
•Interface and communication with ELWIS
•Interface communication K2/ELWIS
• Add/classify the case in K2
•Show the scanned application in K2
•Make controls, create error Messages
•Show decision status
•Send denial letter
•Send case to the queue / next step in the process
•Pick up the case from the queue
•Show the note field
•Create a comment
•Show decision
•Send decision letter

Figure 9: Processes, routines, and system requirements

The blue steps in figure 9 represent three steps in a pension process; receive the application, create a case in the system and handle it manually or automatically, depending on the situation. The white boxes illustrate the routines that should be carried out in the different steps. For example, creating a case in the system means that the administrator (or the computer system if handled automatically) opens the case, checks the details in the application and sends the case to the queue if the application is correct. But if it is incorrect the administrator sends a denial or complement letter to the pension saver. Processes and routines describe how PPM operations and practices are organized and work. The yellow boxes are the system requirements or the system functionality needed to carry out and administrate the routines and the processes. For example, there must be a system functionality to scan the application and communicate the application view to the ELWIS system, then the ELWIS and K2 systems must be programmed to do different things such as showing the scanned application on the screen, make automatic controls, add a status and order number on the case, create a letter, allow the administrator to create a comment. The figure and its content are not exact, these are just short examples what it is about and how these terms were interconnected.

Identifying the processes and elaborating the routines and specifying the system requirements required simultaneous integration of a wide range of competences from different departments and units. The challenge was to integrate the project members' different ideas and knowledge early and make sure that important perspectives had been taken into consideration so that changes late in the process could be avoided. This is further described in the next section.

4.2 Activating a wide assortment of skills

The project's scope was unclear and steadily growing before the restart since the project members constantly identified new needs and new requirements. Due to coordination difficulties and misunderstandings late and costly changes had often been made in the requirements and system functionality. To end this negative spiral the role "Requirement Coordinator" was established, as explained earlier, and her task was to combine people's different knowledge and opinions into a coherent set of processes, routines and system requirement specifications. How should this be accomplished? The project members in the "Operations Subproject", who created the pension and fund processes and identified and specified the IT-system functionality, came from different departments and had different ideas and interests on what the system should be able to do. Moreover, PPM and the project managers emphasized the importance of taking the future users', that is, the pension administrators' opinions on functionality and user-friendliness into account so that the switch from the old system to the new system would be a pleasant experience. Future system users should thus also be involved early in the process and participate in the specification of the system. The Operations Subproject Manager said:

> ”The participants of the 'Operations Project' have very different backgrounds. Some people have a distinct theoretical background and some people work more with concrete practical administration. (…) It is a teamwork to figure out what we need and how it should work at all instances; what is good for a administrator is one thing and for this an administrator's competence is needed and when we must consider the laws a law competence is needed to understand what the law and the regulations say, so there are many competences involved."
>
> (Kristina, Operations Subproject Manager)

The wide spread of knowledge was needed to carry out the project task but knowledge related diversity meant also that people looked upon things from different perspectives which earlier had resulted in communication and collaboration difficulties. The "Operations Subproject" members' diverse background and expertise meant that they were exposed to different kinds of information and used to different ways of working. For example, some project members had a theoretical background while others only worked with practical tasks. Some other people preferred to work with images and icons whereas others were more comfortable with numbers and words. One strategy to avoid many changes late in the process was to try to reveal, collect and integrate the members' different knowledge, perspectives and opinions simultaneously all at once, so that people would not come afterwards and argue that they needed to add something and make changes because they had misunderstood each other or seen that the requirements were incomplete. This would also result in an improved overall view of processes, routines and system functionality needs, according to the Requirement Coordinator and many members of the "Operations project". I identified two work models or interaction forms that aimed at integrating various people's knowledge. These will be presented one by one in the coming two sections.

4.3 Dry land swimming and the creation of a trial product

When the requirements for the telephony parts of the system were created a form of prototype-thinking and fictive simulation, a paper-based representation of a particular section of the system was developed – some sort of a "trial product". The purpose of creating a "trial product" was to facilitate joint understanding and develop a concrete common view of the system and reduce the risk for misunderstandings. The project members argued that it made it easier to imagine how it would be to administrate different pension issues in reality if there were some pictures that represented the screen windows and menus and views to experiment with. One "Operations Subproject" member said:

> "We started to 'dry swim' as we called it, to get an understanding how this would be in real life because it was really difficult to imagine all these windows that would pop up. So what we did was what the IT-people called 'low-tech'. We quickly created paper drawings in power point and used parts from previously created requirements and then we started to experiment with different user scenarios. For example, 'imagine a pension saver calling in saying that he has lost his PIN-code and needs a new one.' Then we experimented; 'now I have this window with this information with name and personal details. Then I need this button or system functionality to order a new PIN-code. And then the outbox square must turn up'. (…) Without the 'dry land swim' we had not be able to visualise and get a common view of what it would be. Everyone would have had his or her own mental picture."
>
> (Jessica, Operations Subproject Member, Subproject Manager Implementation and Data Conversion)

End users, that is, pension and account administrators and the Subproject Manager for Implementation and Data Conversion were the main participants in the work process of elaborating the telephony requirements. According to the Operations Subproject Manager it was important to consider what kind of people one worked with. She assumed that it might have been less necessary with prototypes if there were only very theoretical people who were used to abstract thinking. But she still argued that there is always a risk of misunderstandings anyway since theoretical people may then be in their own world thinking out solutions alone. The Operations Subproject Manager said:

> ”One must think of what people are in the project. If there are three very theoretical persons with good capability of abstract thinking, one might need less prototypes. But I believe it is needed anyway to make sure we say the same things so that it does not turn up later 'oh, is that what you said but that is not what I imagined'. Because then one has been in his own world thinking out things by oneself."

> (Kristina, Operations Subproject Manager)

The project members who were involved in the making of the telephony requirements and the creation of the "trial product" argued that this was a good way of working since it facilitated joint understanding and a common view of what functionality would be needed in the future when the administrators at the call centre should answering the phone and administrate and handle different requests.

Unfortunately, there were some drawbacks with the visualization and prototyping practice. The Requirement Coordinator said that the elaboration and specification of telephony requirements was done without understanding how these parts would interplay with other parts of the system. She still underscored that it was good and important that the end users had participated in the process but that the writing of the system requirements should have been done by professional requirement people because users of a system do not normally understand the system's underlying interactions. She said:

> ”The telephony requirements were carried out by users at the lower levels of the organization. What happened when they drew the pictures was that they had this picture in front of them and tried to include as much information in each picture as possible without understanding the consequences, how it would interplay with the rest. It is here we have had the major integration problems."
>
> (Elisabeth, Requirement Coordinator)

In addition, the requirements turned out to be rather detailed again. That was not the intention from the beginning but it turned out to be so since the participants of this process specified the functionality (often menus, keys, pop-up windows) and placed them on the pictures when experimenting and discussing. Unintentionally, the details of the pictures became the norm for how and where on the screen things should be and what it should look like. The imagination of different administration scenarios became fixed thinking and the design of the screen windows were implicitly taken for granted as the correct and determined versions of the screen views. The "Operations project" members intruded in this way in the designers' domain since the detailed requirements specified not only *what* the system should be able to do but also *how*. Nevertheless, the responsible person for this process argued that the end users were at least deeply involved now in the process, which was important to be sure that the system would meet users' expectations. She believed that if they had not used this work model the users would not accept the system to the same extent since they then would not have had the same chance to understand it and its underlying requirements. The leader of this process explained the situation in the following way:

> ”If you are supposed to keep a certain window in mind and imagine that you should work in this view then you want to have a picture of the view. We started to write in the picture and suddenly things had been fixed; 'up in this corner this bottom shall be placed because everyone has seen it in the picture. The thinking became fixed unintentionally. However, the requirement specifications were still very good and the users were deeply involved in the process. I believe that the users would not have

> understood the requirements so well without the details. But sometimes it turned out to be at the expensive of the designers."
>
> (Jessica, Operations Subproject Member, Subproject Manager Implementation and Data Conversion)

Furthermore, complementing the requirement specifications that mostly were elaborated and documented in dense text form, with paper-based images of the system facilitated and improved collaboration between different members. The work model was especially improved for those who usually worked with icons and illustrations. The leader of telephony requirements had tried to communicate and elaborate the requirements in exact words and detailed texts but misunderstandings occurred time after time. Then when the members started to draw and send images and communicate by marking things in the images misunderstandings disappeared and work pace was speeded up.

To sum up, the "dry land swimming" and the creation of a "trial product" that was based on pictures of screen views reduced misunderstandings and facilitated integration of the project members' skills and viewpoints. Collaboration was improved internally in the "Operation Subproject". Yet, the drawbacks with this work model were that the requirements became too detailed and broke in on the designers' work sphere. As a consequence some system solutions turned out to be difficult to integrate and run effectively. In the next section, we will learn about another, perhaps more useful way of integrating diverse people's skills and perspectives.

4.4 Working polyphonically in cross-functional workshops

During the period before the restart, many "Operations Subproject" members experienced that the work and the meetings were disorganized and confusing as described previously. After the restart that was changed. The meetings and the workshops were better planned in terms of what should be done when and where and who should participate. In this way the participants got a better chance to join and contribute and their perspective and knowledge could better be utilized and integrated in the process. The previous informal, unstructured way of working and communication was thus changed to a more organized, pre-planned interaction which project members appreciated.

Most of the work with the processes, routines and requirements was carried out face-to-face to in workshops. The arrangement of workshops constituted another way of integrating diverse people's knowledge and differed from the "dry land swimming" activity but also here different people convened at the same time to contribute with their various perspectives skills. The way in which the workshops were run and the conversation pattern that emerged during these sessions seemed to lead to early integration of knowledge and multi-dimensional and stable requirements. This is explained in the present section.

The Operations Subproject Manager organized and planned the work and the workshop meetings and invited people to participate and distributed the agenda and handed out material and documents in advance to facilitate participants' preparations for the meetings. The Requirement Coordinator took the role as chairman during the workshop sessions which the other participants thought was logical since she had been working with similar things before. According to the members of the Operations Subproject the Requirement Coordinator was extremely knowledgeable and hard-working person. The Operations Subproject Manager said:

> ”She is not an ordinary person. She has some kind of over-capacity, over-energy which we have wondered where it comes from. She has an enormous ability to keep lots of things in mind at the same time as people come by and interrupt and ask things all the time. She has no set-up time which is very unusual and in addition she has worked extremely hard, she is like 2.5 people.”
>
> (Kristina, Operations Subproject Manager)

The workshops were organized in the following way. The Requirement Coordinator pictured the processes and routines on the whiteboard. She stood up and created the drawings with different boxes and arrows and colours and notes for different parts, steps and routines. The other participants, mostly experts from different departments of the organization, sat down around a table oriented towards the whiteboard and the Requirement Coordinator. The participants brought law books, directories, documents, papers and prior work sheets to be able to investigate things during the session. Different issues of the processes and routines were discussed and the Requirement Coordinator plotted in their contributions in the emerging process description picture on the whiteboard. The picture got more and more details and the project members could see how it was developed and together detect undeveloped spots in the process and routine descriptions and identify where information was missing. According to the project participants, the visualization made it easier for them to see the same thing at the same time and explore the details of the processes, routines and requirements together. They could also see how their knowledge, opinions and comments where put together and how their knowledge complemented were interdependent and interconnected with each other and where opinions diverged. The Requirement Coordinator explained that this was a way for people to understand how one’s own explanations and statements become interpreted and fit (or do not fit) with the others and argued that this kind of workshop was useful when people understood things differently:

> “It is a clear way for people to understand that ‘this is not what I thought’. Because if you talk it is very easy for me to interpret it in my own way and then I keep on interpreting it in my own way and then after three months we realize that we did not mean the same thing at all (…) For example, we had this guy from the law department who had very different opinions compared to those who came from the customer support service. But that is why this form is so good because you can talk and say ‘alright, we view it differently’ but then we at least understand that we understand differently. Everybody is taking part in the process so that no one says that ‘they do not bother what we think and now it ended up like this’. ‘You look at it in this way and it is okay if that happens on your side but we want this because we work like this’. I think people must understand and accept that people reason in different ways in different parts of the organization. Therefore I like this form of working.”
>
> (Elisabeth, Requirement Coordinator)

Moreover, it resulted in that the project members could learn from each other at a general level where their knowledge intersected, how things interplayed, how to solve general problems and carry out this kind of task, and how a new "world" like this could be formed in cross-functional workshops. The Operations Subproject Manager said:

> ”And when we worked with 'Application for Pension' we had these people who did not work with pension at all but then later when we talked about their area they had been involved in and heard the discussion; this works but that does not work'. There are similarities which could be learned and used."
>
> (Kristina, Operations Subproject Manager)

In this way the participants got an overall picture of the organization's processes at the same time as working out the details of the routines and the IT-need. In the following sections we will dissect a particular conversation from a workshop session to see how different people's skills and perspectives were utilized in the work process and integrated in its outcome. Five distinct characteristics of the conversation have been identified which will be discussed in 4.4.1. These characteristics formed a conversation pattern, which is explained and illustrated in 4.4.2.

4.4.1 Exploring details and integrating perspectives in conversation

The creation of the processes, routines and system specifications required integration of a wide range of knowledge. To really understand how this was accomplished in the workshops one must take a closer look into the conversation that unfolded among the participants during the meeting. Five conversation characteristics have been identified. First, people's various perspectives and differences in understanding were revealed in the conversation. Second, details of the problems were elaborated and deepened through comparison with previously solved problems. Third, aspects of the problem that needed more exploration by specialists in the line functions were identified in the conversation. Fourth and most importantly, different people's knowledge and perspectives were blended in conversation in "the same here and now". Fifth, the Requirement Coordinator synthesized what the different participants contributed with and put the comments together and applied the various contributions in proper sequence and time when elaborating the processes and routines. The five characteristics were recurring and the conversation continued to display this pattern until the task or problem was solved. I will show these characteristics one at the time by using short conversation extracts that I picked up from a four hours long workshop session. Before showing the extracts and the characteristics I present the overall topic and explain what the meeting was about.

The workshop concerned the elaboration of a certain process, "Application for Survivor's Pension", and the creation of the routines that must be conducted to carry out this process. The elaboration of a process was a complicated matter that took many days and hours to complete. The conversation extracts that I use here come from one of the "Application for Survivor's Pension" sessions. The participants were; the "Requirement Coordinator", one person from the "Communication Department", one from the "Pension Unit", one person from the "Law Department" and one from "Insurance and Planning". The problem that they had on the agenda for this meeting was to investigate what would happen if the applicant of the "Survivor's Pension" had turned 50 when he or she applied for the "Survivor's Pension" and what that it turn means for the elaboration of the application form and what kind of effect it would have on the administration of the application. The Requirement Coordinator started the workshop discussion by introducing the problem and stated what they needed to do during the session. The Requirement Coordinator had been noticed that there must probably be four squares on the application form that the applicant should mark with a cross, instead of three as was stated earlier. This additional square and cross implied that the process and the routines must be changed. During the first hour of the workshop about 25 different aspects were discussed related to the identification of the "Survivors' pension process" and the specific problem situation, which points at the immense of details and information that these conversations involved. Out of the discussion on the 25 aspects of the "Application for Survivor's Pension" I identified the five characteristics and the communication pattern, mentioned above, but since the conversation is difficult to grasp without a contextual understanding and since I cannot include the whole conversation I will explain what it was about and then show some particular conversation lines to illustrate the characteristics and the conversation pattern. After 15 aspects the first four characteristics had been identified. The fifth characteristic, that is, the conclusion of the discussion, or the synthesis as I called it, was formed by details from the first 15 aspects together with the next ten aspects of the conversation. However it would be too much to include all 25 aspects so I will explain the first 15 aspects shortly, just to let the reader know what the discussion concerned and more easily understand the short extracts from the conversation presented below.

First, the project members discussed the meaning of manual administration and the additional "cross" that should be included in the application form of Survivors' pension, and why the pension saver should fill in the cross if he was over 50 years old and recently have had a child. Then, second, the participants in the meeting also reasoned that the condition put on the pension saver, or *how* the condition was expressed in the pension saver's application form, probably was hard to understand for the applicants. Third, one of the participants raised the question whether the applicant must actively have made a fund choice and if that was something to which PPM needed to create a control step in the process. Fourth, they stated that at one check-point in the process it must be investigated if the pension applicant has an account and if she or he already has a Survivors' pension or has had one previously. Fifth, the participants of the workshop discussed if the pension savers might demand a system function to be able to apply for this kind of pension insurance through an automatic phone channel. Sixth, one of the members initiated an investigation on what check-ups can be made when the pension saver logs in and requests this pension insurance via the Internet. Seventh, they clarified the meaning of the "log in" function in relation to the check-up of account. Eighth, a further exemplification of what controls must be carried out when the application form arrives to PPM was mentioned to be further investigated after the workshop. Ninth, it was discussed that EU-issues always should be administrated manually. Tenth, the participants explored the details of a scenario in which the pension saver has sent more than one application form or made complements to the original application before it has been registered by PPM. Eleventh, a scenario was discussed in which the pension saver dies before the request for survivors' pension reaches the agency. Then, twelfth, the conversation continued by investigating a fictive scenario where the pension saver's address was missing when a decision letter was to be sent out and that PPM must await the pension saver to complement the case and application form. Thirteenth, the participants also addressed the issue whether the Internet and PPM's website as communication channel can stop incorrect application forms. Then, fourteenth, the workshop members discussed if and in what circumstances a decision letter should be sent to the estate of the deceased. And fifteenth, the details of a potential case in which a Survivors' pension applicant dies before a complete application has been submitted were explored.

After this presentation of what the meeting's first hour was about I will show the four conversation characteristics and the pattern that these characteristics formed. We will begin with a short conversation extract that shows how the participants revealed differences in meaning and understanding of the problem and clarified the situation together in the conversation. This characteristic of the conversation is included since it was important to detect difference in meaning early because it is expensive and difficult to correct misunderstandings late in the process. The words and sentences in bold in the text box highlight where the differences turned up and were clarified.

Extract 1 – Revealing differences in understanding

1. Requirement Coordinator (RC): Let us go through what we need to do. We have got an email about the drafts that Siv was working with, the new application forms. They have come to the conclusion that they want **4 crosses** and not 3 as we believed. So this is what the application form looks like. The first square here says 1) I apply for ”Survivor's Pension”. Then it is **2) Fill in if you have turned 50 years old. ”I have turned 50 years old and I have got a child within three months from the day I applied for ”Survivor's Pension Insurance”, is what they want to include.** That is, one fills in the square here, or rather, if one is older than 50 the case should **no matter be administrated manually**.
2. Participant 1: Yes…
3. RC: And this we did not have the last time.
4. Participant 2: **Why should it be handled manually regardless?**
5. RC: **Because one really wants to control and make sure that they have the rights to get the insurance.**
6. Participant 2: Okay
7. RC: **So that we do not just deny the application but let them believe that they have the insurance and then tell that they don't.**
8. Participant 2: Okay
9. RC: So we got something else here [shows in the picture and uses post-it notes] because we said this: we scan the application, we interpret the application, verify. Now we talk about the application form road and then we send the application to ELWIS who creates a handling of this. Alternatively, the application comes in via the Internet and there we said that here we had the social security number, the barcode, notes, beneficiary group and that an individual is desired and there [points and shows in the picture] we **have another group then, 50 years old that we must interpret.**
10. Participant 2: **Does one really have to mark with the cross? I think the sentence is quite difficult.**
11. RC: It reads as follows: ”I have turned 50 years old and I have got a child within three months from the day I applied for ”Survivor's Pension Insurance”. I don't think it was crystal clear…
12. Participant 3: Can I ask one thing, **when interpreting the application, not everyone should fill in and mark with a cross, is it right to interpret…?**
13. Participant 4: **If there is a cross we should interpret it.**
14. Participant 3: Okay
15. Participant 4: **It is the same thing with other interpretation…is there something, we interpret, but if there is nothing so..**
16. Participant 5: **Some of these must exist because if one applies for e-insurance it must exist.**
17. RC: **Well in the interpretation it does not care about that, only in the verification. So there we must have a 'cross 50 years old'** (points at and draws in the picture on the whiteboard). Somehow anyway, because **I happened to be in the middle of the discussion between Helen and Alf and Helen argued that it was good with a cross so that one really sees that 'yes, I have understood this condition about..'**
18. Participant 4: **…if I am older than 50 years old**
19. RC: …**I must have had a child'. Then we do not really care that much about the cross, what we should check is that they really have turned 50 years old.**
20. Participant 2: **Alright, so it should be administrated manually regardless of the cross.**

First, the Requirement Coordinator states that there should be one additional "cross" for people over 50 who within three months from the application day have had a child, on the application form. Or rather she states that if the applicant is over 50 the application form should be administrated and processed manually. One of the participants (2) does not see the reason for why the case should be administrated manually, in line 4, which points at a difference in understanding. The Requirement Coordinator explains and clarifies this in line 5 and 7. The Requirement Coordinator explains in line 7 that PPM wants to ensure that the pension saver knows the conditions for this insurance and seeks to avoid letting the pension saver believe by mistake that he or she has the insurance. She puts the new "x 50" in the process context in line 9 and tells that PPM must "interpret" it. Then the participant (2) wonders why the cross is needed, which reveals another difference in meaning, in line 10, and if the pension saver really has to fill it in. And another participant (3) does not understand the need and meaning of the "interpretation act" in line 12. A fourth participant answers by comparing to the general principle or "normal case" in this situation and argues that if there is a cross, PPM interprets it, in line 13, 15. The Requirement Coordinator clarifies that there is a difference between the two steps of the process "interpretation" and "verification" and that the cross is most important in the verification step but that the cross in the "interpretation" step is important from a user perspective "in order to really see that 'yes, I understand this condition that if I'm over 50, I must have had a child'", in line 17). The Requirement Coordinator and the participant 2 reach the conclusion that the cross itself is not really so important from a PPM perspective, and is not indeed interpreted by PPM, what is important to control is the age of the applicant since it determines whether the case should be administrated manually or automatically, line 19 and 20.

Here we could see how this way of working resulted in that various differences in meaning were revealed and also clarified early in the process before things were set and finished, which perhaps saves time and money as well as interpersonal conflicts and frustration later on in the process.

Next, in extract 2, I will show how the participants explored the problem and deepened their understanding of the problem together in conversation. This was done through comparing a new problem with a situation or problem that had been solved before. The project members used a previous experience from solving another problem as starting point or reference point in the solving of the actual problem. The old solution was compared with the new situation to see if it had similarities and could be understood in the same way. If it did not, the project members learned something new about the problem anyway. The bold lines show the reference and comparison statements.

Extract 2 – Exploring the details of a new problem by comparing it with an old problem

1. Participant1: Then we had this with the cancellation order if there is more than one application.
2. Participant 2:We discussed at the last meeting that we thought **it should be the same as, that is, the same as exists today?**
3. **Participant 1: Pension and Fund**
4. Participant 2: Yes
5. Requirement Coordinator (RC): **The rule there is that if it arrives via the Internet the last one counts and if all has arrived via the letter way it is the last one that counts. If several ways are used, the self-service way counts no matter if that is the last one or not.**
6. Participant 3: But arrival date for the first one counts if the applicant has sent in more than one application?
7. RC: That is, even if one has sent in several applications the arrival date for the first one must be the one that counts, right?
8. Participant 2: Yes because we have then sent the case further and then they might think instead that 'I take my wife instead of the kids but then it is the kids anyway.."
9. Participant 3: **So if you send one letter one day and then it arrives the next day and becomes registered and then you visit the website the day after that, it is the register date of the letter that counts even if we have not had time to scan it yet?**
10. Participant 2: So we can see it then?
11. Participant 3: Or?
12. Participant 2: But if it is not scanned, we cannot see it.
13. (Confused chat)
14. RC: But it is the post mark; they install the scanner on the post mark.
15. Participant 3: So it may be scanned a day later or so.
16. RC: **That is, all pension applications are dated back.**
17. Participant 3: Yes.
18. RC: **These have arrived much earlier.** Did we have something else?
19. Participant 1*:* Moreover, we have this that concerned if the arrival date was before death date.
20. Participant 1: Yes we then had to complement the person - that was the scenario, right?
21. RC: No, generally.
22. RC: Arrival date was before death date.
23. Participant 2: Or wait..
24. Participant 1:Or rather, has not got the decision yet
25. Participant 2:'The application may be accepted by death date if arrival date of the application is before death date'. Yes, it is possible.

The problem is introduced in the first line. The situation is that if the pension saver sends in two application forms, PPM must know which one counts, that is the cancellation order of the applications. In rows 2 and 3 the participants assume that the "Pension and Fund processes" is something that they could use as a starting point to compare with when setting the details of this specific aspect of the "Survivors' pension process". In row 5 the Requirement Coordinator" declares what applies to "Pension and Fund", namely, that the last received application is the one that counts - unless one application has been submitted via the self service system, then it is this one that counts no matter in which order it arrived. In rows 6-16 the project members discuss and reach the conclusion that concerning the "Survivors' pension process" it must be the date of arrival of the first application that counts. This thus diverged from the "Pension process". One of the participants imagines a scenario in row 9, in which a survivors' pension applicant sends in an application by letter and then the following day logs in on the website and sends in a new one. In this case it is the letter that counts even if the letter has only been registered but not scanned and therefore is not yet viable in the system. Again after reasoning together, the Requirement Coordinator compares with how pension applications are managed, in line 16, and says that these become dated back in the scanning process to the date of the postmark, implicitly suggesting that the same thing can be done for "Survivors' pensions applications" as well.

Here the participants solved some details of the problem by comparing with the "Pension process", which they previously had elaborated and specified. In the first comparison moment, row 5-9, they found out that the "Survivors' pension process" differed and so "departed" from, or "falsified" their initial hypothesis on how it perhaps would function. In the second moment, 10-18, the project members compared the situation again with the pension process and found and identified a detail of the "Survivors' pension process" that could be treated in the same way as the "Pension process". The starting point or initial "theory" seemed to offer an idea and something that they could refer to, which in turn seemed to facilitate the "exploration" and identification of the new process.

Next, in extract 3 I will show how the workshop discussions also constituted an arena for detecting things that needed to be further explored and developed in PPM's specialist functions and line departments. In this extract it is shown how the participants identified an issue that should be investigated by law experts in the law department. The words and sentences in black mark how and what and where the issue to be further investigated was identified.

Extract 3 – Identification of issue to be investigated in specialist function

1. Participant 1: **We should think a little bit about the one regarding the 50 years old that comes from the Internet because here I want to make a distinction.**
2. Participant 2: Yes, exactly.
3. Participant 1: …because as soon as they log in to apply it is **important that they get another type** if they have turned 50 so that they understand... **we should make a note on that**.
4. Participant 3: I thought…next week we will leave for a conference, the law unit and **Håkan Nyholm and Mia Halvarsson so I will take the application via the Internet with me and the issue what controls we can do at the application moment and what we need to make a formal written decision on that.**
5. Participant 2**: Yeah what we can stop here**
6. Participant 1**: Yes what we can stop because it would be really good if we could stop as much as possible.**
7. Participant 3: **But as it is written today, it is not very well written, we might ask for a change in the law so it might be good if Håkan...**
8. Participant 4: **The problem is presumably that we cannot make a formal decision on the web**
9. Participant 3: **No.**
10. Participant 2: **So we receive it anyway.**
11. Participant 1: **Yes but at least we check "are you in the system", the first thing and if the person is 50…**
12. Requirement coordinator (RC): **'If you are in the system' is controlled because this [shows in the picture] I must have to log in.**
13. Participant 1: Yes…
14. RC: **And if they are not in the system…**
15. RC and Participants (in unison): **they cannot log in!**
16. Participant 1: That's true. That's good.…
17. Participant 1: **But savings and assets…**
18. Participant 3: **I will write that too.**

Participant 1 proposes that those pension savers who apply for this insurance on the Internet and are more than 50 years old should get another version of the application form immediately at the moment when they apply. She wants to make a notification on that, line 1 and 3. Participant 3 elaborates this suggestion, in line 4, and proposes that she will raise the question of investigating what PPM's rights are to control the applicant and the application form on the Internet when her department, the law department, leaves for a conference.

According to participants 2 and 1, the best scenario would be if PPM could control as much as possible of the details that the pension saver fills in to stop errors and to avoid receiving incorrect application forms, and then be allowed to accept or deny the application, that is, to make a decision on the Internet, lines 5 and 6. However, the law, as it is written, hinders this act. The idea of investigating a potential change in the law turns up, in line 7. Formal decisions cannot be made on the website, which means that incorrect applications are received and processed as well, participants 2, 3, and 4 reasoned, in lines 8-10. In lines 12, 14, 15 the participants came to the conclusion that one important check-up could be made on the Internet – if the applicant exists in the system at all. Participant 1 also wanted to add the control of the applicant's savings, which also was put on the list of issues to be further explored by the law experts, lines 17 and 18.

This is one example of issues that were identified during the workshop and decided to be investigated further in the line departments and units. In this extract the participants explored and solved different details of the problem together but there were also aspects and details that they could not solve at the meeting. A law change and different potential Internet-related controls were to be discussed among experts in the law department. Issues were detected that so constituted new "material" to be elaborated within the specialist functions where necessary skills existed. One may say that the workshop provided an arena where people's different knowledge, perspectives and ideas "collided" (here pension law and rights and Internet technology) with each other and in the intersection important things that must be developed by specialists were identified.

In conversation extract number 4 another prominent conversation characteristic is shown. Here the participants' various perspectives and knowledge are integrated in conversation. The project members used their different expertise and viewed the problem from various perspectives throughout the workshop session. Their knowledge were blended in conversation and integrated in the outcome of the workshop meeting, that is, in the process and routine description. This feature permeated the whole session so what I have done here is that I have used the same three extracts again but this time I show where and how the participants drew on their specialised knowledge and used their different perspectives to analyze and solve the problem. Then I will explain how different contributions and details were integrated by the Requirement Coordinator and put into a *synthesis* and used as a part of the solution to the overall problem, that is, the creation of the "*Application for Survivor's Pension*". But first, the extracts are presented one by one again with a description on how the different knowledge perspectives were used and integrated in conversation. The extracts have got new names and titles. Now they are called Extract 4:1-4:3 "Blending perspectives in conversation". Marked lines and sentences point at the different perspectives that turned up in the discussions and became integrated in the process description.

Extract 4:1 – Blending perspectives in conversation

1. Requirement coordinator (RC): Let us go through what we need to do. We have got an email about the drafts that Siv was working with, the new application forms. They have come to the conclusion that they want 4 crosses and not 3 as we believed. So this is what the application form looks like. The first square here says 1) I apply for ”Survivor’s Pension”. Then it is 2) Fill in if you have turned 50 years old. **”I have turned 50 years old and I have got a child within three months from the day I applied for ”Survivor’s Pension Insurance”,** is what they want to include**.** That is, one fills in the square here, or rather, **if one is older than 50 the case should be administrated manually** irrespectively.
2. Participant 1: Yes…
3. RC: And this we did not have the last time.
4. Participant 2: **Why should it be handled manually irrespectively?**
5. RC: **Because one really wants to control and make sure that they have the rights to get the insurance.**
6. Participant 2: Okay
7. RC: **So that we do not just deny the application but let them believe that they have the insurance and then tell that they don’t.**
8. Participant 2: Okay
9. RC: **So we got something else here [shows in the picture and uses post-it notes] because we said this: we scan the application, we interpret the application, verify. Now we talk about the application form road and then we send the application to ELWIS who creates a handling of this. Alternatively, the application comes in via the Internet and there we said that here we had the social security number, the barcode, notes, beneficiary group and that an individual is desired and there [points and shows in the picture] we have another group then, 50 years old that we must interpret.**
10. Participant 2: **Does one really have to mark with the cross? I think the sentence is quite difficult.**
11. RC: It reads as follows: ”I have turned 50 years old and I have got a child within three months from the day I applied for ”Survivor’s Pension Insurance”. I don’t think it was crystal clear…
12. Participant 3: Can I ask one thing, **when interpreting the application, not everyone should fill in and mark with a cross, is it right to interpret…?**
13. Participant 4: **If there is a cross we should interpret it.**
14. Participant 3: Okay
15. Participant 4: **It is the same thing with other interpretation…is there something, we interpret, but if there is nothing so..**
16. Participant 5: Some of these must exist because if one applies for e-insurance it must exist.
17. RC: **Well in the interpretation it does not care about that, only in the verification. So there we must have a ‘cross 50 years old’** (points at and draws in the picture on the whiteboard). Somehow anyway, **because I happened to be in the middle of the discussion between Helen and Alf and Helen argued that it was good with a cross so that one really sees that ’yes, I have understood this condition about..’**
18. Participant 4: **…if I am older than 50 years old**
19. RC: …**I must have had a child’. Then we do not really care that much about the cross, what we should check is that they really have turned 50 years old.**
20. Participant 2: **Alright, so it should be administrated manually regardless of the cross.**

In Extract 4:1 the idea of introducing the “x 50” in the application form in lines 7, 17, 18 and 19 indicate a *pension saver perspective* since the participants look at the situation from a pension saver point of view. The same perspective is shown in line 1, uttered by the Requirement Coordinator and in 10 where participant 2 makes a comment on *how* the condition on the application form was expressed, which she thought was complicated and inappropriate from a pension saver’s perspective. In addition, an *administration perspective* turned up when discussing why this type of case should be administrated manually and whether the cross should be interpreted or not (ex. lines 1, 4, 12, 13, 15, 20). Moreover, the Requirement Coordinator explained how the information was processed in the system, in line 9, which indicates that there also was a *pension process perspective and system technology perspective* in the conversation. And when the Requirement Coordinator stated that there was a difference between the ”interpretation step” and the ”verification step” in the process, in line

17, the *system technology perspective* turned up again. In line 5 a *law and rule perspective* was integrated in the discussion since the utterance concerned what rights and conditions must be fulfilled to get this type of pension insurance.

In extract 4:2, at least four different perspectives were applied and intertwined in the discussion in order to clarify details and solve the problem. First, there was *a law and rule perspective* - the rules for the verification order was what the discussion here was all about, introduced in line 26 and continued in lines 30, 31, 32, 34, 44, 45, 47 and 50. In addition, a *system technology perspective* turned up in a discussion on the distinction between different communication channels, in line 34. Also the *administration perspective* was visible when the participants discussed the scanner and the postmark in lines 37, 39, 40, 41. The project members imagined what the pension saver might think and do, in lines 33, 34, indicating also that there was a *pension saver perspective* integrated in the discussion.

Extract 4:2 – Blending perspectives in conversation

26. Participant 1: Then we had this with the cancellation order if there is more than one application.
27. Participant 2: We discussed at the last meeting that we thought it should be the same as, that is, the same as exists today?
28. Participant 1: Pension and Fund
29. Participant 2: Yes
30. Requirement Coordinator (RC): The rule there is that if it arrives via the Internet the last one counts and if all has arrived via the letter way it is the last one that counts. If several ways are used, the self-service way counts no matter if that is the last one or not.
31. Participant 3: But arrival date for the first one counts if the applicant has sent in more than one application?
32. RC: That is, even if one has sent in several applications the arrival date for the first one must be the one that counts, right?
33. Participant 2: Yes because we have then sent the case further and then **they might think instead that ’I take my wife instead of the kids but then it is the kids anyway..”**
34. Participant 3: So if **you send one letter one day and then it arrives the next day and becomes registered and then you visit the website the day after that, it is the register date of the letter that counts even if we have not had time to scan it yet?**
35. Participant 2: So we can see it then?
36. Participant 3: Or?
37. Participant 2: But if it is not scanned, we cannot see it.
38. (Confused chat)
39. RC: But it is the post mark, they install the scanner on the post mark.
40. Participant 3: So it may be scanned a day later or so.
41. RC: That is, all pension applications are dated back,
42. Participant 3: Yes.
43. RC: These have arrived much earlier. Did we have something else?
44. Participant 1: Moreover, we have this that concerned if the arrival date was before death date.
45. Participant 1: Yes but we then had to complement the person, it was the scenario, right?
46. RC: No, generally.
47. RC: Arrival date was before death date.
48. Participant 2: Or wait…
49. Participant 1: Or rather, has not got the decision yet
50. Participant 2: ’The application may be accepted by death date if arrival date of the application is before death date’. Yes, it is possible.

In extract number 4:3, three different perspectives were present. First, in line 22 and 23 and 25 and 26 and 35 the project members stated that it was needed to investigate what controls are possible to do on the web and how a formal decision could be made and whether a change in the law was needed, which could be said to constitute a *laws and rules perspective*. Second, in line 19, 21, 29, 30, 32, and 33 where the participants considered the difference between applying on the Internet and using the traditional mail channel, both a *system technology and communication channels and laws and rule perspectives* turned up.

Extract 4:3 – Blending perspectives in conversation

19. Participant 1**: We should think a little bit about the one regarding the 50 years old that comes from the Internet because here I want to make a distinction.**
20. Participant 2: Yes, exactly.
21. Participant 1**: …because as soon as they log in to apply it is important that they get another type if they have turned 50 so that they understand... we should make a note on that.**
22. Participant 3: I thought…next week we will leave for a conference, the law unit and Håkan Nyholm and Mia Halvarsson **so I will take the application via the Internet with me and the issue what controls we can do at the application moment and what we need to make a formal written decision on that.**
23. Participant 2: **Yeah what we can stop here**
24. Participant 1: Yes what we can stop because it would be really good if we could stop as much as possible.
25. Participant 3: **But as it is written today, it is not very well written, we might ask for a change in the law so it might be good if Håkan...**
26. Participant 4: **The problem is presumably that we cannot make a formal decision on the web**
27. Participant 3: No.
28. Participant 2: So we receive it anyway.
29. Participant 1**: Yes but at least we check ”are you in the system”, the first thing and if the person is 50…**
30. Requirement coordinator (RC): **’If you are in the system’ is controlled because this [shows in the picture] I must have to log in.**
31. Participant 1: Yes…
32. RC: **And if they are not in the system…**
33. RC and Participants (in unison): **they cannot log in!**
34. Participant 1: That’s true. That’s good.…
35. Participant 1: **But savings and assets…**
36. Participant 3: I will write that too.

What I have shown now is how various perspectives or “knowledge bases” turned up and were used in the elaboration of the “Survivor’s Pension Process”. Related to this polyphonic characteristic of the conversation there were also a certain type of summary statements made by the Requirement Coordinator that integrated and synthesized pieces of what the participants had said. These statements and syntheses were used to fill the gaps in the pension process description. The Requirement Coordinator showed in the picture on the whiteboard how and where the details of the syntheses elaborated the process description. When the Requirement Coordinator had interrupted the discussion and made these summarizing comments and showed in the picture where the pieces fit, she introduced the next thing to be discussed and so the previous discussion was finished. In this way the summary statements seemed to push the work forward and the synthesis appeared to function as a “process driver”. A new discussion evolved and the participants discussed and explored various details of this problem. After a while, the Requirement Coordinator summarized and drew a new conclusion again and related and integrated some of the details of the new synthesis into the main process description. As in a pattern, the Requirement Coordinator invited the participants to an “open and free” exploration and discussion of each aspect of the problem and then she connected visually and verbally parts of the discussion to the whole picture. The “process drivers” helped the participants to see how far they had come and indirectly made them end discussions on particular things and so continue the process. This seemed important since the spontaneous discussions easy involved lots of information and many details and relationships. In such circumstances it is easy to lose direction and sense of the whole. So the process drivers seemed important to avoid getting lost in the details.

Extract number 5 is an example of a synthesis, a process driver statement made by the Requirement Coordinator. This was uttered within the first hour of the workshop when the first 25 aspects of the problem had been discussed, explained in the beginning of this section, after the conversations that were shortly presented in this section. The synthesis contained details from these conversations and the words and sentences marked with bold are examples of that. While talking the Requirement Coordinator elaborated the picture of the process and the routines on the whiteboard and added the details from the conversation during the first hour, for instance, the controls of the applications that must be made; here it simply was if there existed a case since before that should be complemented by the new application, or if the person already has a “Survivors Pension”. If no case existed a case should be created in the system and here the Requirement Coordinator reminded that the participants concluded that the arrival date for the first application will count. Then she said that if the application is sent in via the Internet the process steps may differ. She recalled that some of the controls may be the same concerning the potential existence of an account and whether the applicant still is alive and then she picked up from the conversation in extract number 3 that savings must be controlled as well and also if there is an EU-case. From conversation extract 2 details regarding death before arrival date was plotted in the process and routine descriptions. This conversation also contained elaboration of the process steps if there are insufficient savings and if the address is missing (not included in the extract).

Extract 5 Synthesis – process driver

So we said that here [shows in the process picture] we **controlled the application** and what we control in this stage is only **if there is a case since before** and then we should complement the issue, or **if it is someone who already has a “Survivors’ Pension**”, then we should not. In this case we said that we create and send a general case, a request case where we say that ‘this is something that we will handle but we do not know what you mean; did you want a change or what?’ and then we must take care of that and make a decision. (From conversations 1 and 2. . See also the introduction to the workshop where the different aspects were presented)

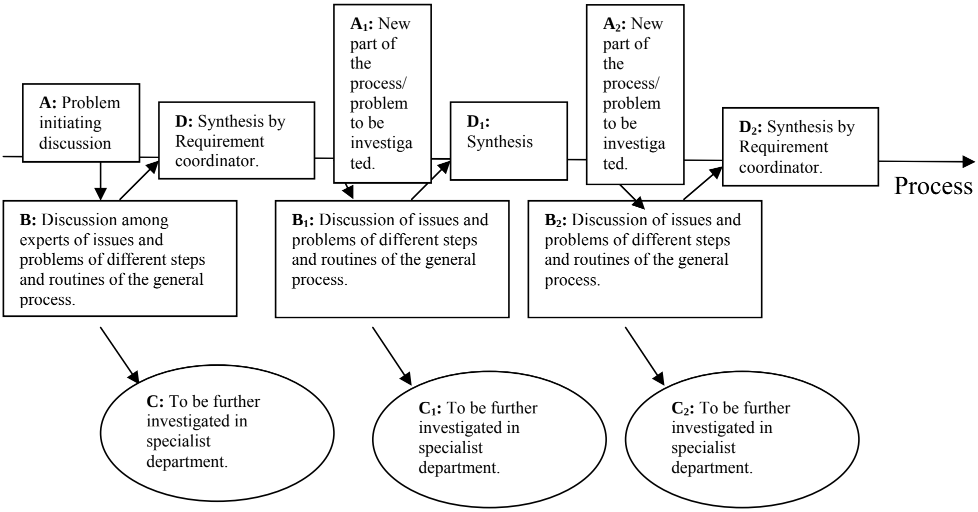
Otherwise if we have not got a case we will **create a case** and **we said that we save arrival date and we wrote that arrival date for the first application will count.** If the **Internet** creates a case one must complement that case and in this situation it may be so that we have sent it further and if so we must end that observation. Then we check the handling again and it is important to control here [shows in the process picture]: if **there is an account, if they are alive,** the same controls as we do here [shows in the process picture] and then we check **the savings** here and we said also that we control if they happened to have an **EU-case**. (From conversation 3. See also the introduction to the workshop where the different aspects were presented.)

Are they about to withdraw the money from the account or do we have money coming in and what could happen here. By the check-point it could go in different directions; **death before arrival date, if insufficient savings we make a denial decision** and create a decision letter and then we close the case. Then we had a question about the issue with the address. (From conversation 2. See also the introduction to the workshop where the different aspects were presented.)

So, the main point with the synthesis and process driver explanation is to show how the Requirement Coordinator in her role as chairman of the workshops led the work process forward with this kind of statements. The synthesis became an element in the conversation pattern. The other characteristics made up the other ingredients in the conversation pattern. In the next section I will illustrate the conversation pattern and show how the conversation characteristics and the synthesis related to each other.

4.4.2 Conversation pattern

The figure below summarizes the preceding section with the different conversation characteristics. Here the conversation features are put together in a model that shows how a typical conversation in the workshops unfolded. The different parts or elements of the process are described in detail in the preceding section. The Requirement Coordinator introduces the problem that the participants are supposed to discuss and solve together (A). A discussion follows (B) among the different project members. The discussion concerns different details of the problem and the different steps and routines of the pension processes. During the discussion the participants reveal differences in understanding and elaborate details of the problem and deepen the knowledge about the problem. Comparing the actual situation and problem with an old problem of similar type that had been solved before was rather common. Most importantly, the participants discussed the problems from various perspectives and integrated in this way their knowledge in conversation and produced an interdisciplinary understanding of each problem. Moreover, during the discussions the project members also detected issues that needed to be further investigated by experts in the line departments (C). Then, when the problem had been discussed and analysed the Requirement Coordinator put details from the conversations together and summarized parts of what the participants had said and formed some sort of synthesis (D). The details of the synthesis were integrated in the general description of the pension process and the routines. When the Requirement Coordinator explained the synthesis people became quiet and listened to her and looked at the emerging picture on the whiteboard. After this the Requirement Coordinator introduced a new problem to be discussed and so the process went on until the problems were solved and the process finished. The pension processes and routines were basically created in this way.



A: Problem initiating discussion
D: Synthesis by Requirement coordinator.
A1: New part of the process/ problem to be investigated.
D1: Synthesis
A2: New part of the process/ problem to be investigated.
D2: Synthesis by Requirement coordinator.
Process
B: Discussion among experts of issues and problems of different steps and routines of the general process.
B1: Discussion of issues and problems of different steps and routines of the general process.
B2: Discussion of issues and problems of different steps and routines of the general process.
C: To be further investigated in specialist department.
C1: To be further investigated in specialist department.
C2: To be further investigated in specialist department.

Figure 10: Conversation pattern in the workshops

To conclude, one may say that the cross-functional workshop model was useful since it made it possible for the project members to reveal their differences in meaning in an early phase of the project and it facilitated a deeper understanding of the problem since many different perspectives turned up at the same time and were incorporated in the elaboration of the processes and routines and requirements. The cross-functional workshop was also valuable since it facilitated detection of issues that needed further exploration. And most importantly it facilitated early integration of knowledge, which would reduce the risk of amendments late in the process. In addition, the chairman role implied that the work moved forward and that the participants could see the whole at the same time as engaging in the details. Except for the Requirement Coordinator and the Operations Subproject Manager there were no clear roles in this subproject; the Operations Subproject members coordinated their work only through the joint task of developing the "World of the Premium Pension".

As a next step, the Requirement Coordinator wrote down the system requirements out of the material created in the workshops. These were then reviewed by the members of the Operations Subproject. The reviewing activity became also an important means to reduce the amount of changes in requirement specifications. This procedure is described in the following section.

4.5 Anchoring and stabilizing the goal

The new organization and the new project directives involved that the Requirement Coordinator should write down the system requirements, based on the material that the project members had been working out during the workshop sessions, and make sure that the Operations Subproject members agreed on the requirements so that changes could be avoided further down the process. The requirements should be written in accordance with certain PPM templates and different versions should be logically dated and numbered. The Operations Subproject members met and went through each line in each requirement and discussed and made changes together. This is the last part of the first leg "From vision to specs".

The review of requirements should be carried out formally and be documented and signed. It was a "theoretical" work, which again complicated the integration of knowledge and the collective imagination of what the outcome would be. The Operations Subproject Manager said:

> ”The reviewing was very theoretical which was difficult since one then must translate the requirements into a practical solution and imagine what it would be like in reality when administrating the case. And the gap between theory and practice is huge. The requirements specifications were immense and the rules and regulations were complicated. And there was only text like 'if this happens, that must happen but if any of this happens these letters must be sent and if these letters are sent the letters should have this information'. There were many details and discussions like 'is it really so, we must check it in the regulations' and somebody goes through the pension law and tries to interpret it in detail. The administration of a case can go in different directions at all levels and during the reviewing meetings one is supposed to carry all circumstances and details in mind and imagine what it would be like when administrating the case. It's difficult."
>
> (Kristina, Operations Subproject Manager)

When the requirements had been reviewed and accepted the members should sign to show that they had mutually agreed on the requirement documents. There were no department managers or board members involved in this activity. Furthermore, there was an enormous time pressure so the time for reading the requirements documents before the meeting was sometimes close to zero, according to the project members. Also during the meeting the members had to work very fast. The Operations Subproject Manager said that "the tempo was horrible and one just had to try to keep up with the speed". Moreover, at this stage in the process there were only guesses and hypotheses; nothing existed yet in reality so the project members did not know for sure if every need and situation was covered. It was difficult to imagine what the end result would be; what the whole system actually would look like and how it would feel to work in the system and if it surely would contain everything that was needed. One could not be perfectly sure that no crucial details or functionality was missed out. The uncertainty depended on the fact that the project was extremely big, according to the project members, even after the reduction in functionality and system requirements, and on the fact that it was created completely from scratch. The Requirement Coordinator said:

> ”The most difficult phase in the project, from my role's point of view was when everything still only existed in theory. We had an enormous amount of requirements documents but nothing to look at. (..) This is exceptionally big, when one carries out minor changes it is easier because then one has a framework as a starting point. We had nothing so we had no chance to see how it would work in reality."
>
> (Elisabeth, Requirement Coordinator)

However, the review process continued in the same way throughout the project in spite of the difficulties with the way of working. There was too little time to change work model and reflect over the way of working and if there could have been a better way to review the requirements, the Operations Subproject Manager said. Nevertheless, since the requirement process before the restart had been confusing for both those who worked with the requirements and for developers, the reviewing procedure and the contract among "Operations Subproject" members was yet a way to make the process more stable at least. Many different people from the organization's various departments and units (no managers) were involved in the review process, which should enhance the quality of the requirements and reduce the risk that something important was missing or described incorrectly. The members now agreed on what the system should be able to do and in this way the process was "anchored" and more fixed even if some changes would be necessary further down the process. The project goal had so been stabilized, which would make the situation easier for designers and developers in the following phase since it now became clearer what the developers were supposed to deliver. No more changes in requirements made by the "Operations project" members were permitted after the reviewing process. In addition, the designers were also involved in the review process to raise quality in the requirements and ensure early that the requirements were possible to solve. Illogical, too complicated and expensive as well as uncertain requirements were so immediately taken away or changed. Thus in this time-pressured situation the members had no better choice than invite as much and as different people as possible to review the requirements and together create a "qualified guess" how the pension processes, routines and the computer system would be organized and work in the future.

The complete system requirements were consequently sent to the designers who gave opinions before the requirements were finally settled and "frozen". Negotiations between the designers and the requirement people occurred since the designers wanted to avoid unreasonable needs from a technical perspective and the requirement members wanted to avoid inadequate solutions from a user perspective. The discussions between "Operations project" members and designers implied that there was some overlap between the first steps in the project's waterfall model, that is, the requirement step and the design step. The intention was to ensure that the "Operations Subproject" members did not order system solutions that were too expensive or time-consuming or impossible to build. The Design leader said:

> "Designing is about creating a solution to the requirements and integrating the different IT-systems in order to solve the requirements. John and I received the requirements and discussed them at the overall level quite early before we elaborated the details of the system solution to say what was possible and not; 'we cannot do this and we cannot do that' because otherwise it easily turns out to be too expensive and complicated.'"
>
> (Simon, Design Leader)

The designers became more involved in the requirement phase than previously. The management's choice of time as steering parameter helped the designers in their discussions with the Requirement Coordinator in the sense that she must take the developers' opinions regarding time and complexity into considerations when formulating the requirements. The Design leader stated:

> "When there was no time pressure the "Operations members" simply said; 'just build it!' Then she [the Requirement Coordinator] has changed her attitude very much during the journey and listened more and more to us."
>
> (Simon, Design Leader)

Moreover, the routines that also demanded system support to be carried out could no longer determine the system's construction, that is, *how* the system should fulfil these needs. The requirements should only involve process descriptions which also resulted in that the designers and developers got more freedom and decision rights in the creation of the system solution. When the designers and the Requirement Coordinator and the "Operations Subproject" members had agreed on both the system requirements and the design solution and when the documents had been formally "frozen", the next phase could begin, that is, to turn the design drawings into a concrete computer system.

5. The second leg – from drawing to device

The intention with the waterfall model and the new design step in the developing process was to improve communication and collaboration between "Operations project members" and "IT-project members" and to early ensure that requirements could be turned into a usable computer system. In addition, the creation of design drawings and documents should also improve collaboration internally among IT-project members and integration of system solutions. It had previously been difficult to coordinate actions and solutions due to lack of structure and integrative framework that kept things together. However, in spite of the promising idea of creating a system design at two levels (overall and detailed design), it turned out that some developers did not accept the idea and the new work model, which caused serious problems in the IT-project. This section's first headline 5.1 "On collision course" will report on how the IT-project failed again due to work model and collaboration troubles, weak leadership and cross-functional misunderstandings between developers and testers. In 5.2 the measures taken to get out of the new crisis are described. These measures implied that interaction and communication changed and improved, which is explained 5.3. Eventually, it seemed that the IT-project members had succeeded to produce high-quality solutions that could tolerate fierce inspection and examination by the testers, which is described in 5.4.

The second leg, from drawing to device, was carried out from the summer 2003 until the fall 2004. During fall 2004 most functionality had been developed and programmed and the developers were mainly occupied with bugs and error correction. One thing that is important to know is that the requirement specifications were divided in three isolated parts, blocks or as the project members called it "delivery packages". These packages were created and delivered to the IT-subproject members one at the time. This implied that the developers could start working on the first package before the requirements in delivery package number two were perfectly completed. When the first block of functionality had been programmed it was sent to the testers and the developers could start working on the next delivery while the testers were examining package number one. In this way especially testers did not have to sit and wait as long as if all requirements and all details in the system functionality should be completed first before the testing could commence. The waterfall model should be used for each package.

5.1 On collision course

As mentioned above, it seemed that some developers did not like the new waterfall model and did not follow the project directives implemented by the project management. Developers' strong individual will and old habits proved to be harder to change than expected. In combination with weak technical leadership this resulted in that tensions in the IT-groups started to grow during fall 2003. Moreover, in November 2003 the first package of programmed system solutions was delivered to the testers, which turned out to be quite a failure too since the testers and the developers had not agreed on the specifications of the content or the method of delivery. In addition, the quality of the system solutions was inferior and a new crisis became a fact. The new "direction" that the "ship" took after the restart unfortunately appeared to be a "collision course". This is described under the subheadings 5.1.1-5.1.3.

5.1.1 Low acceptance of the new work model

During fall 2003 it turned out that some developers did not work in accordance with the new method and the waterfall principles. They were expected to create design sheets (overall and detailed) before programming but this new design step had not been fully accepted by the developers and incorporated in their work procedure. The overall design should be elaborated first and thereafter the details of the design and then the coding should start. But it turned out that sometimes the detailed design was created before the overall design, which can be compared to drawing the details of a room of a house before the architecture of the house in its whole has been created. Some of them neglected the detailed design and continued coding the solutions before the detailed design was specified and documented and approved by the design group. Often no detailed design was elaborated at all, which meant that developers started programming "right out of their heads" without creating or following any drawings at all, as was common before the restart. One developer said:

> "We do not work by the book; some people don't follow the method. Overall design and detailed design should be made before programming! There is no technical leadership."
>
> (ELWIS Developer)

The development process with its different sequential steps had thus not been accepted by all project team members. This made it difficult to keep pieces together and to coordinate actions and maintain a coherent overall view of the project process. It provoked tensions within the group since some developers tried to adopt and follow the new principles and thus considered the use of different work methods frustrating and confusing. It also provoked tensions between developers and managers who tried to control the process and plan the work. Why did not all developers want to work in accordance with the new model and principles? At least four reasons for that were explained at an IT "reflection" meeting.

First, some developers asserted that the new method decreased developers' possibility to invent and program creative technical solutions. The design leader explained that the new work method implied that the design group set the frames and limited developers' space for coming up with own innovative ideas. That is, the designers made the drawings that technically specified what the developers should build. The new development process implied thus less room for developers' creativity and autonomy. The design document implied that the developers were given an "order" to build something already fixed and determined. The design details that the developers should specify involved too little creative freedom in the developers' view. In addition, the developers' suggestions on the details must be approved and accepted by the designers, which implied that the developers' creative autonomy was reduced even more. This became a tedious way of working for the developers. Although the designers enjoyed the design work they could admit it was a waste of talent since the developers also were experts. The design leader stated:

> "We sort of stuffed their throat with different things and said 'build this!' They are skilled people whose competence is not being fully utilized."
>
> (Simon, Design Leader)

Second, the detailed design step in the new waterfall model implied that the developers now had to document the details of the system solutions and write down how they intended to solve the problem. This was a big change in the work procedure and the rules; before the developers just solved the problem. Project managers argued that developers often thought that this was time-consuming, boring and even unnecessary. One of the Programming Leaders explained some developers' attitude in the following words:

> "Everything that is not technology is shit."
>
> (Jack, ELWIS Programming Leader)

In the designers' view the detailed design was a way to communicate that developers and designers as well as the Requirement Coordinator looked upon the problem in the same way, that both the overall design and the details matched and solved the underlying requirement. The Design Leader said:

> "If we have got a requirement I need to know that the developers have understood the problem and intend to deliver a solution to this problem only, and not something else".
>
> (Simon, Design Leader)

However, the developers who created the detailed design solutions had poor understanding of the underlying requirement specifications since they never reviewed or saw the requirements. This will take us to the *third* reason behind the low acceptance of the work model. The developers did only get the design documents but did not know the reasoning behind them, the purpose with them or how the solutions would be used in the future, that is, what operational problem the system solutions would solve. As a consequence, the solutions often mismatched the requirements and the technical system construction. Elaborating the detailed design document and the overall design separately resulted in a *technical gap* between the two design levels. The designers and the developers argued that the detailed solutions were not as good in terms of quality and inventiveness as they could have been if they had collaborated and created the detailed design and the overall design together.

Fourth, there was also a *time gap* between the handover of the overall design documents and the beginning of the work with the details. One of the programming leaders explained that when the design sheets were handed over to the developers the designers wanted that the details of the solution were elaborated right away so that the problem could be solved in its whole. However, the developers were often overloaded with work and could seldom receive and take on the work right after the designers had finished the first design part. The "waterfall" did not flow smoothly. The time gap resulted hence in that the designers missed early feedback on how the details would fit with the whole. The designers did not know if the developers had understood the problem since they did not get any feedback or confirmation on how the details could be developed within the general design set. Some of the developers were able to begin the detailed design and discuss with the designers in time but that was not enough since nothing could be programmed and implemented before all details were set and approved. The time between handing over an overall design document and the start-up of the work with the details was several weeks and even a couple of months sometimes. This delay implied that a written description and documentation of the details became even more important, which some of the developers disliked and often ignored as described above. The designers still argued though that the developers must follow this rule since the developers in a way then, by showing in text how their design details would fit in with the whole, confirmed that they had understood the overall design and the requirements.

Taken together, these circumstances resulted in that the developers did not appreciate the new work situation. In addition to the low acceptance of the new work principles, the developers also had individual habits and strong wills, which will be explained in the following section.

5.1.2 Individual habits, own agendas and lack of technical leadership

The Head Project Manager said that the problems in the IT-project did not only depend on the inherent difficulties with the work model itself but with the fact that the developers had a “strong will, old habits, and own agendas” and brought it up as a serious problem at several board meetings. Even though the Head Project Manager had talked to the developers and explained the idea behind the new work model the developers still continued working as they were used to and did not adhere to the process steps or the rules. At one of the board meetings the Head Project Manager said:

> ”I have talked to them and they assure that they understand the idea with the method. They nod in the right place but then they do as is in their bones anyway. It is an anarchistic gang with own agendas.”
>
> (Thomas, Head Project Manager)

The developers constituted a diverse group of around 10 different strong individual characters. They were cultural carriers of different technical schools and had different ideas, opinions, habits, styles and personalities. The age varied also with approximately 25 years. The developers were all devoted to their work but everyone wanted to do as he always had done and believed was most appropriate or constituted the best method or most innovative and elegant solution. Developers commonly sought technical perfection. One IT Line Manager, as well as some developers, stated that developers are like artists – they require artistic freedom.

Moreover, the steering parameter “time” was also something that some developers did not pay much attention to. Some developers still valued technology advancements more and strived for perfect solutions. Even after a problem had been solved someone could still be working on it to implement a better solution. One of the Programming Leaders stated:

> ”They wanted to invent something that was 3 % better if they could. Some of them did not care about the time plan; they wanted to produce good and fun solutions. Time plan and such things – that’s someone else’s problems.”
>
> (Jack, ELWIS Programming Leader)

The problems with the work method and the developers’ strong habits and will often resulted in long informal discussions on how to do different things and which solution was best. Discussions were endless. The work was disorganized and solutions were still difficult to integrate. Several IT-project members argued that the ELWIS developers were in need of a stronger technical leadership. One developer said:

> ”Everyone thinks he knows best. No one gives up and no one can end the discussions.”
>
> (ELWIS Developer)

The developers and designers said that there was no technical leadership in spite of the fact that there were both a Programming Leader in the ELWIS group and an IT-subproject leader.

They argued that the IT Subproject Manager was a good planner and administrator but that she did not have the same technical skills of this specific domain as the developers and therefore she could not exert much influence on the work. The Design leader asserted that the reason why the developers needed a manager was to have someone to discuss technical problems and solutions with, someone who could help them make wise decisions and lead the group in the same direction. The IT Subproject Manager said that she could not take part in the developers' discussions or control the situation and the work. She said:

> "If they argue that they need a certain amount of time to finish something – that's how it will be. I cannot argue against them. I cannot reach them."
>
> (Martina, IT Subproject Manager)

The Head Project Manager was frustrated during the fall since he did not get enough information about the problems in the IT-groups. He received signals during several months that the IT Subproject Manager was not the right person to lead the ELWIS group. She could not manage the developers and there were too many technical and organizational problems outside her area of expertise. The Head Project Manager started to attend the IT-project's status meetings since he wanted to come closer to the developers and get a better insight into the communication and collaboration in the group and how the work was carried out. The Head Project Manager said:

> "I did not know what was going on. I did not get a feeling for how things went on, only that it did not work. The developers said that they lack technical leadership. Martina did not have the situation under control."
>
> (Thomas, Head Project Manager)

The low acceptance of the work model, the individualists in the IT-group and the lack of technical leadership were more than needed to make the project members experience a new time period of uncertainty, conflicts and slow progress. To make it even worse, it turned up that the delivery of program code to testers constituted a complicated and problematic area. The system solutions in package one proved also to be of bad quality. The following section describes this. All these problems resulted in a new crisis, which required new powerful interventions and dramatic changes to be solved.

5.1.3 Delivery conflict and inferior system quality

In November 2003 time had come to deliver the first package of program code to the test group. Beside the importance of keeping the time table, the first delivery was also a critical and prestigious event since the developers and the whole project now should prove that they were capable of creating high-quality system solutions in time; prove the strength of the new organization and the project members' competence. A Monday morning in the middle of November 2003, the IT Subproject's Manager and Programming Leaders convened as they usually did on Monday mornings to report on how the work progressed. Unusually however, was that the Head Project Manager and the project owner (Insurance Department Manager) also attended the meeting. The top managers wanted to hear and see with their own ears and eyes what the developers said and get a first-hand feeling for the actual situation and the developers' chances to deliver the first package on time. The IT Subproject Manager went through the time plan’s activities and asked each programming leader what tasks and activities could mark as finished and how much work was to be done on the other tasks. Then the IT Subproject Manager asked “will we deliver on Friday?” and the programming leaders said “yes, as it seems today”. There was still quite much work to do so the project owner wished them good luck and said that she looked forward to the end of the week. There was at this point in time no discussion about any problems or risks. And as developers had prepared and planned, code was delivered to testers almost in time. However, what happened was that the testers got frustrated and angry since the delivery did not meet their expectations regarding content, quality, instructions, information or documentation. And most of all, the testers criticized the bad quality in the code. The testers argued that this depended on that the developers had not confirmed that each component functioned well, that is, that the developers had not carried out component tests before deliverance which the developers should have done to verify that each program component had been properly programmed.

The developers asserted though that some component tests had been done, otherwise one does not even know if the problem has been solved or not. However, the developers had not had enough time to perform more robust and comprehensive tests. No hours were specifically calculated for tests in the developers’ time plan. One of the programming leaders reasoned however that the testers have both time and competence and could easily perform those tests. The Test group and the ELWIS group had different opinions on what and how much should be tested by the developers before delivering the code to the test group. The testers argued there was a certain point in keeping the groups separated because the testers could then test more objectively how different code or groups of code were created and worked together. The developers argued that it did not matter who carried out the tests since both developers and testers all belonged to the same project now, “we are in the same boat”, one of the programming leaders said. The Head Project Manager said that he was surprised that there was no routine for this established in the organization since before - it could not be the first time that the developers delivered code to the testers. There is no “delivery culture”, as the Head Project Manager called it. The testers and the developers explained that this discussion had turned up many times before and that there was no solution to the problem.

The problems with the work method, the diversity in the IT-groups and the lack of technical leadership, the delivery conflict and the inferior quality of the system code together required new changes in the project organization and in the way of working. It is described in the next section how managerial interventions together with bottom-up solutions changed the project’s collision course and saved it from sinking again.

5.2 Managerial interventions and bottom-up solutions

The ignorance of the new method, the diversity of the group, the passion for perfect solutions and the weak leadership resulted in a culture crash with many endless discussions and sharp tensions in the IT-project group. For several months the IT-work suffered from indecisiveness and conflicts in the team. In addition, the conflict between developers and testers made the situation even more difficult and infectious and the quality of the program code was inferior. Consequently, the first period after the restart resulted in a crisis. In the beginning of 2004 the developers made a significant change in the sequential work model and the Head Project Manager implemented some organizational changes, which together took the project in a completely different direction. This is described in 5.2.1-5.2.4

5.2.1 Changing IT-leaders and members

The first and most dramatic measure was that one key developer in the ELWIS group had to quit. His consultant contract was terminated. The reason behind the decision was to make the group easier to manage and to end all different discussions that impeded project progress. Some IT-project members argued that this measure was absolutely necessary in order to take control over the group and calm down the situation. The design leader stated:

> "It was perfectly correct to take Dennis out in this situation and show that now 'this is serious'."
>
> (Simon, Design Leader)

However, this action made some people in the project upset and offended since he was considered one of the best programmers and was important for some other developers' work. Also, he was not the only one to blame for the crisis in the group, some IT-members argued. Now he became stamped as the black sheep. One of his closest colleagues reacted by not continuing the work almost at all for several weeks. According to one of the programming leaders he interpreted the action as a signal that all the work that they had done together was bad. The IT Subproject Manager said that the Head Project Manager made the wrong decision. This was the most knowledgeable programmer and he was extremely important for his closest colleague. The Head Project Manager admitted that the programmer was a skilled person but that he was at the wrong place at the moment. He said:

> "Dennis is a talented programmer but he is completely wrong for this group right now. Tom reacted very strongly but I have talked to him several times."
>
> (Thomas, Head Project Manager)

In January 2004, 8 months after the restart, also the IT Subproject Manager was "kicked out" as she expressed it. She argued that the Head Project Manager had used an authoritarian leadership style during the whole fall and exemplified with that he had attended her meetings with the developers, decided on details, said no to breakfast meetings and coffee breaks with bakery and sweets. However, the design leader and the other programming leaders reasoned that changing IT Subproject Manager was necessary since the IT Subproject Manager was more of a project planner and administrator than a technical leader. The developers argued that she did not have the appropriate skills to guide and manage the design and programming work or discuss and solve the organizational and technical problems. Both the manager and the developers underscored that there were no personal conflicts involved but stated that the professional side of the communication did not work. Several IT-managers in the line organization were asked to step in as IT Subproject Manager but no one accepted the offer. The Head Project Manager interpreted it as a fear of failure since it at this time still was unsure if it would be easier to manage the ELWIS group or whether the project would deliver the computer system on time and succeed or not. The Head Project Manager must take the role as IT Subproject Manager himself and had so two roles from January 2004 until the end of the project. The Design leader stated that the Head Project Manager understood the technical problems sufficiently compared to the previous manager:

> "Martina could not control the group. She did not understand what we were talking about. She was like a fuzzy filter against Thomas so he did not know what was going on either. When we

> talk to him directly we leave out the most insane technical details, since he is not a technician either, but he understands the problems. If we have a boss who does not understand the problems, we don't need to talk at all."
>
> (Simon, Design Leader)

A third change in the IT-team was the recruitment of a new ELWIS Programming Leader, Jack. He came from the line organization's IT department. The previous ELWIS Programming Leader would be on parental leave for several months and then come back to the group as a programmer without technical leadership responsibility. The other programming leaders, some developers and various members from different parts of the project argued that the new Programming Leader managed to improve the situation in the ELWIS group. According to the Head Project Manager the new Programming Leader became a key success factor.

> "It is very much because of Jack that this was possible to complete."
>
> (Thomas, Head Project Manager)

The Programming Leader of K2 said that the new ELWIS Programming Leader understood the technology and that he therefore could discuss problems and solutions. He was also able to estimate time to do different things.

> "Jack understands and knows the technology. He has an own feeling for different things. To Martina one may say 'it takes 8 hours' and then the discussion is over."
>
> (Ben, "Programming Leader K2")

What seemed important was that he made decisions when needed and took responsibility for the consequences. His success was described by various people in terms of his sense of accountability and that he did not accept requirements that he considered too optimistic without demanding more resources (or time). The ELWIS Programming Leader said:

> "I take the responsibility if something goes wrong. But if I got more work to do I also must have more time. I do not accept to get more requirements to implement but not anymore time. That is an equation that they must learn cannot be solved."
>
> (Jack, "Programming Leader ELWIS")

He also made choices and decisions even when diverging opinions seemed equally valid and possible. In this way he ended discussions and showed the group that he took the technical leadership even though the developers still were the technical experts in the project; the developers had the deepest domain knowledge. According to the Head Project Manager, Jack was empathetic and respected the developers' differences and increased the pressure on them gradually. Jack also described the developers as a group of individualists. He said that when he first came in there were big "ego wars" in the group and that one of his greatest challenges in the beginning was to create trust among the developers. As an example, he explained that the developers created individual "time space" for them selves. They did not report immediately when a task was finished but pretended that they still were working on something. This saved time to work on something else or improve a thing a little bit extra, or just do something funnier, irrespective of what the plan said or what was agreed on earlier. The "ELWIS Programming Leader" argued that to be able to lead the developers one must first learn how they prefer to work and what mentality each one has and then meet and treat them individually. He said:

> "There were big 'ego wars' in the beginning. They created "time spaces" for themselves and did not report to me immediately when they had finished their tasks. One must get to know them. Tom needs to be constantly given new tasks and be occupied all the time; otherwise he flies away and comes back after a week or so with something that he thinks is super cool. Carl's outbursts are just something that one has to withstand. If there only is a couple per day – it is good. Some people prefer to work alone and that must be accepted too."
>
> (Jack, "ELWIS Programming Leader")

To further calm down the situation, reduce the number of discussions and make it easier to lead the work, the whole group of developers should not be involved in all tasks. Instead they should work more individually, peer-wise or in small teams and be assigned a limited task and responsibility. If a problem turned up that could not be easily solved Jack discussed it with the Head Project Manager and the other programming leaders. His usual way of acting was to first "collect" and put the other people's opinions against each other, and then he summarized and structured them into a couple of alternatives and repeated the consequences of each alternative in terms of time, effort and functional performance and argued that a choice and decision must be made and that both could not be done and to do nothing was not an alternative.

5.2.2 The priority list

Moreover, to further make it easier to govern the ELWIS group and make technical decisions, the project management elaborated a "priority list", which specified what was most important to do, second most important and so on. The time factor was deeply emphasized in this document. Quick solutions should be preferred to time-consuming high quality solutions. There had been many discussions in the group concerning quality of the system and "non-functional requirements" such as performance, robustness, maintenance. These software aspects took much time to discuss since people had different opinions on the issues and even if the quality and performance must be considered it was not stated in any document, handbook or guideline what was acceptable standard or who should have the responsibility and right to decide. It also takes much time to build and implement quality in the system architecture and in the solutions and these hours had not been calculated in the developers' time plan in detail. The priority list should from now on shorten discussions and coordinate actions and thus support the new ELWIS Programming Leader in his management role.

5.2.3 "Police" inspections

The Configuration manager or Quality Controller at the project office, the "project police" was also instructed to increase the presence at various meetings and take an active part in discussions to make sure that the work was executed by the book. Routines, rules and procedures should be followed strictly. The project police asserted that "some bureaucracy makes people think". He assumed that people needed the rules to think carefully how to carry out a task instead of just doing as one was used to, at least if new rules and routines have been implemented to break different non-working routines and individual habits. The "police" cautiously and powerfully controlled the development process, the agreements and the way things where delivered. The project office members functioned as a "living tollgate" when requirements and system code where handed over between the groups. The project police even showed up and took an active part in ad-hoc meetings in order to assure that the routines, project priorities and the method handbook were followed. Some project members said that the "police" had sometimes been a "nightmare" but at the same time extremely necessary to coordinate the project activities and lead the project members in the same direction.

5.2.4 Creating the overall design and detailed design together

At the same time as these managerial interventions were implemented another more "bottom-up" driven change was made. Instead of developing the overall design first and separately from the detailed design, the developers and designers decided to create these two design documents together and simultaneously. This implied that the space for developers' ideas was enlarged and the developers' knowledge and ideas were taken care of and incorporated in the design solutions. It facilitated coordination between designers and developers and resulted in a more integrated solution. In this way the problem with the time gap was solved too. This meant that two of the steps in the waterfall model were changed and merged into one step. The "Design Leader" said:

> "First we carried out the overall design and the detailed design separately but then we changed it and created them together simultaneously. Big brush touches and small brush touches at the same time. It kept things together. It was much better. It resulted in improved solutions. This collaboration worked out very well. There were talented people."
>
> (Simon, Design Leader)

The changes altogether resulted in work improvements and a better atmosphere during spring 2004. Even if the developers were given more creative freedom in the design work, most changes resulted in that spontaneous individual action was hindered. The group dynamic was softened and the group became easier to manage and the leadership grew stronger and clearer. Time was still the most important steering parameter and it was even more emphasised in the priority list ("good enough" quick solutions should be preferred to time-consuming ones). There was less room for developers' individual ambitions and intentions. Their most extreme ideas were silenced. The developers had to live with imperfections and "good enough" solutions. Nevertheless, discussions on what "good enough" really meant occurred every once in a while. The design leader said:

> "The group calmed down and it became easier to lead them with the priority list and without Dennis. The technical leadership improved and became clearer owing to both Jack and Thomas."
>
> (Simon, Design Leader)

At the same time as the changes of leaders and members in the team took effect also the interaction and communication in the IT-project changed. Previously, the formal meetings had been concentrated around the time plan and how each activity in the plan progressed. The meetings now contained more and more problem discussions and fewer time estimations and time plan conversations. Most interestingly in the new conversation pattern was that the Head Project Manager and the developers and designers utilized emotional markers and metaphors to analyze problems and facilitate understanding and make common decisions on technical issues. This is explained in the next section.

5.3 From status update meetings to problem-oriented inquiry sessions

As a contrast to the weekly status meetings in the IT-project during the fall 2003 where the time plan was controlled and updated the meetings in 2004 were more focused on discussing problems. The programming and design leaders were supposed to hand in a short written status report before the weekly meeting to the Head Project Manager. This report should include a list of finished activities, present work, actual problems and current critical issues. The minutes of the meeting were saved and summarized in a formalized document.

Instead of going through several detailed planning sheets the meeting was now organized in a "round the table" manner. That is, the programming and design leaders were one by one given some time to discuss current problems. The developers shot in spontaneously to discuss each others' problems but some order was maintained by the Head Project Manager and the project "police". In the search for a solution the developers collectively reasoned and used and helped each other. It seemed that if somebody was asked to freely ventilate his or her current problems it also awakened and got the other persons involved. The problem orientation and "open communication" seemed thus to put the other person's "knowing" in motion as well. The meetings in 2004 involved thus a stronger mutual commitment to each others' problems.

In these problem-oriented meetings there were no strict communication rules but the Head Project Manager was in charge and could distribute the word among the participants, push a decision and stop discussions. The Head Project Manager did not know the answer to the different problems or did not always have an own idea how to deal with the problems but was sufficiently knowledgeable in technical issues to raise questions, structure the alternatives, summarise the situation and put forward a conclusion. In a Socrates-fashion he did not always need to make the decision himself; it was created by the answers to his questions and comments. The questions seemed to focus and narrow the space of the mind of the developers since they must answer the questions and could not therefore engage in too many potential alternatives and consequences. Together the Programming Leaders and the Head Project Manager tried to focus on the core of the problem and limit possibilities in order to make a decision, rather than extending the problem to include *all* potential circumstances and occurrences. Progressing in this phase was about to analyze what gives the highest value and quality within the given timeframes.

Generally, the IT-meetings in 2004, when the planning logic was set aside, displayed more collective reasoning in the conversations and the role of argument, explanation, definitions, contradiction and negotiation was more dominant compared to the meetings in 2003. The conversation can be seen as a rational inquiry process characterized by utterances that for example defined, categorized, evaluated, explained, elaborated, refined, criticized, falsified, clarified, and justified the problem and its potential solutions. In addition to this "fact-based rational argumentative" style of the conversation there was also another more emotional or non-factual aspect of the conversation associated with intuition, values, metaphorical speech, analogical reasoning, judgement and feelings. Emotional expressions of this kind appeared to facilitate joint problem solving, mutual understanding and decision making. This is explained in the next section.

5.3.1 Emotional dimension in conversation

As mentioned previously, the IT-project members argued that the Head Project Manager understood the developers' problems even though he was not a technical expert or had the same domain knowledge as the developers and designers had. The design leader explained that when the developers discussed problems with him the technical details of the problems were left out to make him understand the most important parts of the problems. The design leader said that the Head Project Manager understood what was needed to understand in order to make a decision and help the developers solve a problem. But if the Head Project Manager did not understand the technical details of the problems, on what did he base his decision, what did he listen to or what did he look after? It is here the emotional side of the conversation comes in. The Head Project Manager and also often the Programming Leader of ELWIS frequently asked the developers not only about the logical reasons and explanations to different problems but also how different developers felt about different problems and solutions. The managers listened to the developers' worries, appraisal of different situations and personal judgements and hence used emotional signals to in a way act upon other people's intuition and inner knowledge. The managers also often communicated their own emotional evaluation of different situations to facilitate understanding and collective problem solving. The Operations Subproject Manager said for instance:

> "When Jack came in, one got to know more about the situation in the IT-project; one got a feeling for the problems and how things proceeded."
>
> (Kristina, Operations Subproject Manager)

In order to explain how the emotional dimension operated and facilitated communication among diverse people and speeded up problem solving I present five examples from five different conversations and meetings and show where and how emotional expressions turned up and were used. The five conversation extracts show the same thing but I have chosen to keep them all anyway. It is an attempt to be clear and persuasive in showing something which might be a little unexpected and difficult to explain and observe. Emotional expressions and personal evaluations are highlighted in the extracts with bold marks.

Extract 1 – Emotional dimension of conversation

1. HPM: What do we say about the framework for access rights?
2. EPL: There are some decisions that we have to make about which environments this should be activated on. It is problematic since the test tool makes different things it should not in order to generate data. Either we have to go through all test cases of the test tool, all of them, or we have to make a decision like “okay, we do not run the framework for the access rights in the development environment or the application test environment, while in the integration environment it could be switched on.
3. **HPM: Yeah. Does it feel alright?**
4. **DL: It feels like a risk, I think, if you do not have it switched on.**
5. HPM: Yes there is a risk.
6. DL: There is a difference, I think, with application tests. That is, if you click around in K2 using the web then it has to be switched on.
7. EPL: What environment are you talking about?
8. DL: Regardless of the environment when testing those things. When testing functionality, that is application tests, it has to be switched on, otherwise you do not test.
9. EPL: Yes but then everybody has to go through all the test cases.
10. DL: They should run the test tool in another environment, do you understand?
11. EPL: No.
12. HPM: The problem…I see what you mean, we should run K2 tests and ELWIS tests in two different environments. K2 should have had the framework of access rights switched on, you say. ELWIS tests are dependent on the test tool. We should make a decision what to do.

The first excerpt is from a typical “rational” problem conversation that also contains emotional markers and personal judgments. I will first explain the “traditional” or rational character of the conversation and then point at the more emotional-like side of the discussion that I argue facilitated communication and mutual understanding.

The conversation is about a technical problem regarding safety and the framework for access rights and its effects on the test tool for component testing and the test environment. The framework for access rights does not work properly with the test tool. The conversation starts with a question uttered by the Head Project Manager (HPM) in line 1 directed towards the ELWIS Programming Leader (EPL). The Design Leader (DL) is as involved in the discussion as the Programming Leader. The EPL states in line 2 that a decision must be made and defines and explains how he views the problem. The DL adds his view and categorizes the situation in line 6 and clarifies his own statement. In line 7 the EPL asks for a clarification and in the answer in line 8 the DL gives a definition of test. In line 9 the EPL implicitly takes the time aspect into consideration since it takes time to go through all the test cases. The HPM tries to paraphrase the problem in line 12 and emphasizes the need to make a decision. This is an open discussion where critical questions are continuously asked which all demand “rational” explanations.

However, the Head Project Manager asked the developers already in the beginning of the conversation in line 3, right after the first problem description had been stated by the Programming Leader in line 2, how the problem situation and possible solutions felt. The Head Project Manager could not judge or make a choice based on the first information since he did not have sufficient insights of the technical problem and its consequences. He yet needed to make sure that a quick and appropriate decision was made so that the development process could continue. Instead of asking for more technical details directly the Head Project Manager asked them for an emotional evaluation of the situation. In this way central aspects of the problem came up, in addition to an emotional evaluation and personal judgment in line 4, which resulted in that the Head Project Manager soon understood the situation, in row 12.

Extract 2 – Emotional dimension of conversation

1. HPM: So you have found a new fault?
2. D1: Diane had just received and sent it to the technicians but had not got the update yet but they work hard on it.
3. **HPM: How critical do we asses it?**
4. **D2: Quite critical. It gives Adam some problems.**
5. (D1 and D2) *Discussing the technical details of the problem and conclude that there are ways to get around the problem.*
6. **D1: It is actually easy to fix.**
7. D2: Yes, it sounds like a bug.
8. **HPM: Okay, what is the next step, how do we move on?** Wait for them until next week?
9. (DI, D2, D3) Yes. [Continue and discuss other problems.]
10. **HPM: It feels good.**
11. **D3: That's my feeling too. After the meeting last Monday it did not feel very well. [..] But now it seems optimistic and promising to the 7th.**
12. HPM: It's good that we have not found anything new that does not work; **it feels as if we are in control now at least.**

In Extract 2 the Head Project Manager (HPM) and three developers (D1, D2, D3) discuss a problem that had turned up in the telephony development earlier the same morning. The Head Project Manager asks about the status, how critical the problem is at the moment in line 3, that is, he asks for a personal evaluation or judgment, instead of for example asking about the details of the problems. One of the developers says it is quite critical in line 4 because it has negative consequences for another developer (Adam). The question results also in that the developers start to discuss the problem with each other for some minutes, line 5, and that discussion leads to a slightly different outcome and also another feeling because the developers realize that the fault is quite easy to fix, line 6. The Head Project Manager does not have to worry and wants to move on to discuss other current problems, line 8. Here the Head Project Manager saved them all time and effort since instead of asking about technical details that would probably have required some time to explain and understand to reach the same conclusion (that they do not have to worry about it and should better move on with something else) he asked for the developers' evaluation and feelings towards the situation and problem. The developers discuss a couple of other problems while the Head Project Manager is listening. In the end the Head Project Manager says in line 10 and 12 that it feels good and that things seemed to be under control, which one developer confirms in line 11.

Extract 3 - Emotional dimension of conversation

1. EPL: We have ordered the format, XML export format.
2. DD: What do you mean 'ordered', did you design it too?
3. Design leader: Yes, decided the format.
4. **EPL: And Carl has estimated the time to implement the format and get an XML to a week. And then, since Carl ended his sentence about that it will take a week by asserting that 'this is useless', I ask WHY? [...] What makes me worried is when Carl, when it is Carl with his intelligence who says that something is a problem. Then it is serious.**

Let us move on to the conversation in extract 3. The ELWIS Programming Leader (EPL) had previously asked one of his developers (Carl, Database Expert Developer, DD) to make a time estimation to implement a certain technical tool. The developer had said one week but ended his sentence that the tool will be useless. This made the Programming Leader worried since he deeply respected the developer's competence and intelligence, which he tells the participants of the meeting in line 4. Since the ELWIS Programming Leader did not have sufficient knowledge about the issue he had listened carefully to this emotional expression and evaluation to make the “right” decision in this situation. The developer’s personal evaluation resulted in several long discussions with many different people since the problem behind this utterance was complex and consisted of several parts and perspectives that needed to be investigated before a decision could be made.

Extract 4 - Emotional dimension of conversation

HPM: On the IT-side things go on quite well too. **It is the framework for access rights that haunts us; I don’t feel satisfied with it.** Nevertheless, there will be a delivery on Wednesday as planned with the right functionality and decent quality. Genesys has delivered the interface and some other things. **There was a problem but how it was solved I do not know but Adam seemed satisfied.**

Excerpt 4 is just one sentence said by the Head Project Manager (HPM) in a project meeting with the other subproject managers. This shows how the head project manger uses emotional or value-laden words in his communication to the other subproject managers about how things proceed (first part of the sentence). He shows also that he trusts the developer which he refers to, Adam, and Adam’s feeling or judgment of “Genesys” interface components that had been received. The Head Project Manager said that he was not involved and did not know how some particular problems had been solved but Adam seemed satisfied, which was most important at the moment. The Head Project Manager had thus listened or looked for Adam’s personal reaction and was content with that Adam seemed pleased with the result and did therefore not dig deeper into details, problems and solutions.

Extract 5 - Emotional dimension of conversation

1. **EPL: Basically, this is it. We shall finish all this during the coming week. When it comes to fund change I am not worried. Concerning the transaction robustization, I believe it is clear and easy for Simon and Carl.**
2. **HPM: How does it feel?**
3. **DL: Well, we tried something out last Friday that did not go very well.**
4. **HPM: What does it mean? That it can take indefinitely much time...?**
5. **DL: No, we know what the solution is and we know what we must do. We just have to make it work in practice too. It is the implementation and to make it as good as possible. So I am not worried.**

Episode 5 is a typical conversation between the Head Project Manager, the Design Leader (DL) and the ELWIS Programming Leader (EPL). In line 1 the ELWIS Programming Leader reports what is to do the coming week and that he is not worried about it, which might be said to be an emotional and personal evaluation of situation. The Head Project Manager (HPM) asks the Design Leader how the task feels that he is up to, in line 2. The Design Leader answers that one thing that had been tried out did not work very well, in line 3. The Head Project Manager asks in line 4 what it means in time, line 4. The important time aspect is thus also present in the conversation to be kept under control. And the Design Leader explains the problem at an overall level (no technical details) and concludes that there are no worries, in line 5. Here the participants exchange evaluations, judgments and emotions in every line.

To sum up, the managers and the developers did not perhaps think it was useful to learn in detail about each others’ problems or maybe there was not always enough time to go through all details and possible scenarios and alternatives. Communicating emotional evaluations seemed often to save time and effort. In addition, metaphors and analogies were also used in conversations to facilitate understanding of different problems and collectively find a solution and way ahead. When fact-based reasoning failed metaphors and analogies and emotions seemed to support the members in “untying knots” and move forward in the problem solving processes, especially if the problems were complex and relevant knowledge distributed among different group members. This is described in the following section.

5.3.2 Metaphors and analogies in conversation

Metaphors and analogies constituted the second interesting conversation characteristic during spring 2004. Metaphors and analogies were used by the IT-project members to explain viewpoints and make arguments more convincing and the content of the conversations simpler and easier to understand. In some situations, the metaphors and analogies helped the participants see the same thing. At other times, metaphors facilitated mutual understanding even though the developers still had their individual opinion. To this section I have first brought one short conversation extract to introduce what it is all about in a simple way. Then in the conversation in excerpt 2 a longer conversation is presented, which contains many different metaphors and analogies – almost in every line. However, in reality the metaphors and analogies were not expressed and dropped in the conversation in every sentence but I had to shorten the conversation and cut away lines that said the same thing and parts that did not contain metaphors and could be excluded without destroying the “flow” of the discussion. This means that the sentences in extract 2 did not follow exactly as they are presented here since lines in between have been taken away. The content and the storyline (in which order things were said) have been kept intact. The point is just to make the conversation characteristic of using metaphors and analogies as clear as possible without forcing the reader to go through dozens of pages of conversation transcriptions. Marked sentences and words denote where the metaphors and analogies turned up in the conversations.

Extract 1 - Metaphors and analogies in conversation

1. HPM: It is clear that we should not do it in the 5.0 but the question is if we shall squeeze it in the 5.1?
2. QC: Since it requires 30 hours testing, it feels completely out of the question.
3. RC: I do not even believe this is necessary.
4. (Simon) I do not understand what we should do with it.
5. QC: I interpret it as a residue.
6. RC: I am a little bit hesitant to these things. I think we can put it on the residue list and see if it turns out to be a big job. If everyone thinks it is tiresome to get an error message...
7. **DL: It is a flaw. It is the same category as the ’log outs’ that are placed a little too much to the left.**
8. HPM: Alright, then we agree on this one. Now this is finished then.

Extract 1 is the first example that shows how a metaphor and analogy helped the Head Project Manager to understand a problem and make a decision. In this extract the participants, the Head Project Manager (HPM), the Design Leader (DL), the Quality Controller (Q) and the Requirement Coordinator (RC), discuss the importance of a fault that has turned up in the tests. The importance of the problem is crucial to estimate since not every bug should be corrected, only the most important ones. The Head Project Manager states in line 1 that the bug is not critical but wonders whether it should be included in the last delivery. The Requirement Coordinator argues that it can be postponed until after the implementation to see if the users complain and assert it is a big problem (line 6). The Design Leader states in line 7 that this is only a *flaw*, which is a metaphor to say that it has no functional value at all, and uses a similarity (it is the same thing as having the log out bottom slightly more to the left on the screen view than was agreed and required) to explain why. This ended the analysis discussion and it was decided that no time should be spent on this issue, line 8. Here the metaphor and the analogy facilitated joint understanding and helped the project members and the Head Project Manager to make a time efficient decision, which was important since time saving alternatives were prioritised in this project.

Extract 2 - Metaphors and analogies in conversation

1. DD: Can I tell what I think is wrong. We can start from here. The XML format is wrong. It is not very readable unless we create a **reading tool**. Then to find the changes and understand what it means is almost impossible.
2. DL: **It is like reading a 'word document' without 'word'.** It is not very easy. And it becomes even more difficult because it is binary. **There is a gap between....**
3. **EPL: ...the data that comes out and the interpretation of the data**
4. DL: But to start with, the format is okay.
5. DE: I believe we are discussing different things here. The mission was not that we should create a complete solution to present this, if I understood the report correctly. We shall develop the file where one can…
6. DL: ... detect the differences somehow.
7. DL: If one cannot count with **getting a report in the head** that says that these are the differences on these places, then one has to start from the beginning 'control 1; do these look similar? Yes. Ok, check. Control 2; do these look similar? Yes. Ok, check.' It is a manual work without tools. But step 1 is to create a format that works. In step 2 or so we improve it, if we notice that this is not enough, it does not work, one cannot work in this way.
8. **DD: It will be like comparing the first and the second edition of a book and trying to identify the differences.**
9. DL: More or less.
10. DD: It won't be easy.
11. DL: **Step one is to get a raw format.**
12. EPL: Do you have a better idea?
13. DD: No, I do not have a suggestion because what we do now is not wrong but we stop before we can use it.
14. **DL: Yes, it is right, we all agree on that, but our time is limited now. We do not create a Rolls Royce, only a Fiat right now, to which we will attach chrome strips in phase 2.**
15. **EPL: Okay, so this is the good enough solution with which we can with a dog's job synchronize two databases? Do we agree on that?**
16. DL: We cannot synchronize but compare two, which is the purpose now, but one must deal with the problems manually.
17. EPL: It is the difference tool that puts up the limits.
18. **DL: DD wants a helicopter tool and that is what I want to, everyone wants that, but it takes time to create the helicopter tool. I believe that if we just create the format we can give it to somebody in the street if we are in a hurry. But we must decide the format first. Later if we want a superduper tool we can order it from somebody. But we have not achieved anything yet, I understand the frustration, but we must take one step at the time if we agree on that we are moving in the right direction and not straight into the forest.**
19. DD: I said it was useless which I base on the fact that I had a goal to be able to compare and use it during the delivery the coming fall and that won't be possible if we do not do more and elaborate it further.
20. DL: Sure, there is a risk you are right but it is not useless at any rate, regardless how you turn the reasoning. It might not be enough with the time we spend on it now but it is anyway in **the right direction; we must go here first and then we can take the next step. It is a question on what we do in leg one right now.**
21. EPL: The gist: I still have my worries, DD has the same opinion as before, DL has his opinion too but we agree on that we deliver the first part.

Extract 2 is a technical discussion that concerns the development of a format that should present the content of two large information files so that the files can be compared in order to detect differences between them. Four experts participate in this discussion; the ELWIS Programming Leader (EPL), one Database Expert/Developer from ELWIS (DD), the Design Leader (DL) and a Design Expert (DE). The participants discuss what exactly should be done, that is, what the task consists of, potential solution, and the practical value of the suggested solution. The outcome and the usefulness of the solution is uncertain and the Database expert believes it will be useless and the designers think it will be useful to some extent and something that they can develop further in the future but that it will not be perfect solution. The designers argue that they need to take this first step now and that it will not be worthless since the solution probably can be used for now and then when there is more time the solution can be improved. The participants have different view of the problem and the task and the solution. Discussing the technical details alone does not help them understand each others' view or reach a conclusion. Different analogies and metaphors are used to facilitate the discussion. For example in line 1 the Database Expert Developer states that he will need a tool to be able to read the files if using this format. The Design Leader confirms and explicates in line 2 by using the analogy "it is like reading a word document without the word program". It is a "gap" between generating the data and interpreting the data (lines 2 and 3) However, the task did not include the development of a "reading tool" or "difference tool" to more easily analyze and interpret the differences in the information files, according to the Design Expert in line 5. Hence the question whether the developers will be able to conduct such analysis remains and the Design Leader gives a suggestion how to do it in practice if such analysis report does not metaphorically "fall from the skies right in one's head" (7). The Designer argues in line 7 that it will be a manual work without tools but that step number one must be to create a valid format.

The Database Expert Developer criticizes the solution and states that it will be as "comparing the first and second edition of a book and find the differences" (line 8), which is an analogy that the participants can refer to and agree on, more or less; the Design Leader argues that it will not be that hard to find the differences in the databases but that this is a "raw format" (11) and the first step in a two-step process. The information will be generated by the file format so it will be there in front of them but it will take time to interpret the files due to the enormous amount of data that the files will contain. The Database Expert Developer wants to put more time into it and create a more useful solution to make the information readable (13). The Design Leader argues metaphorically that there is no time to create a "Rolls Royce" now, only a Fiat, which can be equipped with chrome strips in phase two (14). The ELWIS Programming Leader wonders if this solution is good enough even though the work of synchronizing two data bases will be a "dog's job" (15). The Design Leader corrects him and says that it is not possible to synchronize, just compare - the problems must be resolved manually (16). The ELWIS Programming Leader understands that the difference analysis tool will set the limits (17). The Database Expert Developer argues that the solution will be useless unless they do not elaborate and work more on the difference tool. He wants to encapsulate information to obtain a "helicopter view", which the design leader understands and admits would be great in line 18 but that it takes too much time to create now. When the format is created "anyone on the street" can be asked to analyze the files (18); it does not necessarily have to be a project member. But the format must be decided on first. Then the tool can be improved and developed. One step at the time the Design Leader suggests. The most important thing for now is that the developers agree on that they are "on the right course" and not "towards the bushes". This metaphorical argument was an attempt to explain that what was done now would not be useless or in vain even if it might need to be developed in the future. The developers must agree that this is the first step in the right direction (18). The Database Expert Developer asserts that he said it would be useless since he planned to use it for this project's coming deliveries but he will not be able to do that now if the solution is not improved (19). The Design Leader states that it will not be useless since this metaphorically "first step" must be done anyway (20). The participants understand the situation similarly in the end even though different opinions still exist on how much and what should be done at the moment. The ELWIS Programming Leader concludes that he is still worried, The Database Expert Developer has still the same opinion as before the discussion and the designers has also the same view but at least the participants can agree on that the first part should be developed an delivered (21).

In a later discussion with the Head Project Manager and the Quality Controller the Design Leader and the Programming Leader and the Database Expert Developer reported and explained metaphorically that it would be possible to identify differences with the suggested format but that it is a "detective's work" which will take considerable time because it must be done manually. A tool to save time will be needed in the future that is better than "eyes, ears and mouth" the Design Leader argued, like a "browser" he explained to analyse and interpret the data generated by this file format. The Head Project Manager understands the situation and confirms that they for now should work with step one and wait for a future plan that perhaps involves the development of the improved tool. Meanwhile, the ELWIS Programming Leader assures that transition to new system versions will be quality-assured and guaranteed by the processes and routines and system versions controls in use.

To summarize the section, metaphors and analogies were used frequently in the discussions among the IT-project members. This way of talking seemed to facilitate explanation of viewpoints and creation of *mutual* understanding (even if the developers did not reach *common* understanding) and time-efficient decision-making. Moreover, as described in the previous section, project members carefully listened and searched for rational arguments as well as people's worries, intuitions, emotions towards different options, problems, action alternatives. Progress was achieved not solely by letting the best alternative and argument win or by juxtaposing and weaving in many different perspectives as in the requirement specification work but by making time-efficient decisions based on both rational arguments and emotional factors and metaphors and analogies. These conversation characteristics changed the interaction and communication pattern but seemed more common in the early phases of the development process (spring 2004) than in the end (fall 2004) though, which signals that the emotional dimension in conversation and the metaphors and analogies are most helpful in complex situations characterized by much unsettledness, ambiguity and widely distributed knowledge among many different experts.

5.4 Ready to be scrutinized?

The organizational changes implemented in the IT-subproject in the beginning of 2004 resulted in that the collaboration improved. The time aspect became a natural factor to take into consideration in most decisions, at least among the leaders. Project directives were usually followed even though the project members occasionally went off road. Generally, technical IT-problems were solved more peacefully during 2004 compared to the time after the restart in May 2003 until the beginning of 2004.

In addition to the three large delivery packages of functionality, there were many smaller handovers of program code to the testers. These deliveries consisted often of corrected bugs and errors. The project management and the testers wanted the deliveries to be as formal as possible in order to keep control over the errors, different test environments and actual system version for each environment. System code was delivered via the project office and its Quality Controller or Configuration Manager, “the project police”. The testers were satisfied because from now on the testers always knew when and what to expect of a delivery, in terms of content documentation and system version.

As the project process continued and the new delivery procedure became an established practice the developers and testers also managed to go away from the routine and occasionally agree on, between the two of them, to send and receive minor “black deliveries”. This became important in certain situations since the testers sometimes could not wait for the formal big delivery; from time to time they needed a correct version of a function right away to be able to effectively continue the test process. Even though the black deliveries were more informal the developers and testers did not lose control over the situation because the developers knew what the testers needed to know regarding installation instructions and version information and the testers understood what the developers needed to know about the bugs and the delivery request. It seemed that the formal method and process how to deliver things had become their natural way of acting and communicating. That is, the development method and its bureaucracy had been implemented and incorporated in the way of acting as a solid cornerstone from which they could depart without getting disorientated. The way back was also easy to find and in this way the overall sense of direction could be maintained. The quality of the system was enhanced. According to the Head Project Manager the improvement was partly due to the changes described above and partly or perhaps mostly due to a new sort of peer-collaboration between a new test person and the ELWIS Programming Leader. This will be described in the coming section. The improvement of quality in the system code implied that the testers could start running the tests more intensively. And so the third leg “from check to launch” was finally commenced.

6. The third leg – from check to launch

In accordance with the waterfall model the test process started when system code had been programmed and delivered to the testers. The first challenge was to find a solution to the collaboration and quality problem regarding the component testing. How this was solved is described in 6.1. After this first test stage the next type of tests should be commenced, the application testing. Time was running out and the testers were forced to break the waterfall “law” and start application and system integration testing at the same time. In addition, acceptance testing had to be commenced before the integration testing had been completed. This caused a lot of coordination and collaboration troubles and challenges, explained in 6.2. How the work eventually was settled and completed is explained in 6.3.

6.1 Turning a heated conflict into powerful interaction

The first package of functionality was delivered to the test group in November 2003. In line with the sequential way of working the test group’s work load increased with the delivery and now the testers also became strongly involved in the project’s mission.

At the time for the first delivery a big problem related to the way the program code was delivered turned up in the project, as mentioned previously. The problem concerned the imprecision around packaging, specifications of content, installation instructions, preparations and the quality of the code. The main problem was the quality of the code delivered.

According to the test group and PPM's test process, the developers should but had not conducted component tests before deliverance. The developers had run short of time and were delayed and had not set up component testing as an activity in the project plan. There had thus not been time to perform formal component tests to the extent as the testers had expected. The developers argued however that some basic tests had been carried out since that is a natural part of the coding work – otherwise a developer cannot know if the component really has been programmed and finished or not. The testers did not agree and argued that if the developers had done the component tests sufficiently the developers would have detected and corrected things that now were detected by the testers. Now the quality of the system components were so bad so that it was impossible for the testers to carry out meaningful application and integration tests the testers asserted. This resulted in that the quality of the program code was inferior, which in turn implied that the test group could not start testing larger groups of functionality and whole applications as had been planned and expected. In this way the testers would be delayed already from the start which would put the whole project plan and the possibility to deliver the complete system on time at risk.

The developers argued that it did not matter who conducted the tests, as long as a technically skillful person worked on it. The testers did not want to execute component tests since it in their view belonged to the developers' task. The testers should receive finished and completed functionality to be able to start with application testing and then continue with integration testing and performance testing. For objectivity reasons they did not want to assist the developers in the early testing activities. In accordance with the PPM-way (prepared, well-scripted, rational, sequential test methodology) of performing tests the testers understood their role as an unbiased party that should "see with new eyes" with certain distance on the different programs they were testing. The developers pointed at the fact that testers and developers now belonged to the same project and were committed to the same goal. The developers assumed that the testers have had more time than the developers so far and therefore the testers could better perform more rigorous testing. But the test group emphasized the importance of keeping component tests separate from other tests aiming at more objectively examine the system solution. This was formally stated in PPM:s test process. One expert tester explained that the problem is that PPM:s IT-department's test unit has set up a test method that the project wants to depart from:

> ”The line organization wants one thing and the project another concerning testing. In the line organization there is a test process, from which the project wants to depart. The problem concerns component testing."
>
> (Jesper, Test Expert)

The developers wanted one of the best testers to come over to the developer's team to do the tests but the testers refused the idea. The testers said that the problem belonged to the developers and should therefore be solved by the developers. It is not a test group problem but a problem that developers and the project management must solve and how the developers choose to solve it is not testers' business. The test group saw themselves as a line unit rather than a part of the project. The testers basically worked in the same constellation as before and were located in the same place so they did not really feel that they belonged to the project organization. If a tester from the test group carries out the component tests for this time, the developers and testers would end up in the same discussion again when it is time for the next delivery, the testers argued. The testers declared that this is a question of principle.

The Head Project Manager was shocked that this problem had turned since he had expected that there already was an established and accepted routine in place for deliveries since PPM had developed and delivered and tested software code before. Both groups explained that this was not a new problem. The issue was infected and hindered collaboration between testers and developers and hence also further project progress. The Head Project Manager discussed the problem at a project board meeting but no decision was made by the board members.

A lively discussion among developers and testers followed during several days after the first delivery in November 2003. The discussion regarded both the definitions of different tests since the developers argued that they had tested the quality sufficiently before delivery and where the work of the test group actually should begin and how to solve the problem right now. The Head Project Manager attended one of the heated meetings where developers and testers discussed the problem. By asking questions, refining details, paraphrasing utterances and summarising the perspectives and aspects of the problem the Head Project Manager managed to calm down the situation and the participants could together more clearly define the different tests and also most importantly identify and see that there was one test type missing. A slightly different kind of test was needed to be invented and implemented to solve the problem. This new kind of test could be seen as a middle step between component and application tests and is not a standard software testing type. The developers and the testers formulated a name for this kind of test and called it "mini-flow testing". In this way no one was completely right or wrong; the developers had certainly tested, at least to some extent, what they argued was needed to be tested at the component level. On the other hand, the testers were still unable to perform application tests of the functionality since the quality of the components, at least the components' interactions, was inferior. Now with the new "mini-flow tests" small or limited flows of information and processes (and not only static pieces of code) should be tested already at the component level in order to improve the quality.

The second part of the problem concerned who should be assigned the mini-flow testing. The Head Project Manager kept a project perspective on the issue and realized that the project would have to pay for the resource and that the task was crucial and must be solved immediately. He understood the testers' perspective as a client-receiver approach but for him it did not matter who performed the tests, only that the tests were conducted by someone. The developers' schedule was more than over-booked and the testers would have more and more obligations to fulfil as the project proceeded. Since one of the testers knew a qualified person that would be available soon, the Head Project Manager decided to head hunt and pair her with the ELWIS Programming Leader. In the meantime the test group lent a resource to assist the developers in the testing. However, the developers wanted to keep the test group's resource and explained that it would not be possible to introduce a new person to perform these kinds of tests. But the Head Project Manager insisted on this solution and afterwards he maintained that it was a good decision since the pair collaboration between the new resource and the Programming Leader became a turning point for the project and a key to success:

> "Jack and Alice together, not perhaps as persons, but as a function, turned ELWIS from catastrophe to a work of the highest quality"
>
> (Thomas, Head Project Manager)

Also the Test Subproject Manager said that this was a necessary decision. She was a temporary consultant and argued, as many other project consultants, that the discussion on who should do what, whether the test group should lend their resources or not, was a problem of status and prestige at PPM. She said that if the Head Project Manager had not solved the problem in this way now the testers would a year later still be finding errors and writing error reports.

What was it then in the collaboration that was so thriving? According to the Programming Leader as well as the Head Project Manager, it depended on the new resource, Alice's technical knowledge and experience and ambition to understand the system from within. She was not only interested in the "face" of the system but also in its "inner organs". To be able to create this kind of tests she needed to be co-localised and work closely with the system experts and learn and examine the system from inside. Only when one knows the core pieces intelligent test cases can be written to truly investigate the system, several developers argued. According to Jack and other developers a good tester comes close to the system and does not keep a mental or physical distance to it. A close collaboration with the developers is essential to understand the thinking and logic behind the code, the developers assumed. One crucial aspect of testing a computer program is to think of everything that could go wrong and figure out how to create tests that trigger the program to collapse in all those potential ways. It is also about finding any logical inconsistencies in the "hypotheses or theory" put in the requirements and design documents. The tester tried hard to prove that the solution did not work or match the design and requirement specifications by running tests based on the requirements and the design specifications. In this case no prepared scripts or written test cases were given to her since it was a new product under development with new features and interdependencies that they had not started to think about yet from a test perspective. She must experiment and learn from the experiments to understand how to find critical defections. The Head Project Manager explained that the success also depended on the Programming Leader who introduced, explained and showed the system patiently, carefully and pedagogically and so let the new person in behind the "walls". Not only expected faults were detected but also hard-to-imagine incidents and things that absolutely never should happen were found by the test cases that Alice and Jack created together. The new tester learned about the program, executed tests, saved test cases and documented what had been tested and the test result. Corrections of errors were controlled by regression testing to observe that the problem did not turn up again and that no new faults turned up as a consequence of the correction. Without this information it was hard for them to actually know when something was finished. The collaboration was successful and the quality of the system improved significantly. The developers could from now on deliver code to the test group that was of good quality. The code had been reworked and refined through the new sort of testing once already before it reached the testers. One of the most experienced testers also acknowledged the difference and the positive changes that occurred from the deliveries in late 2003 and beginning of 2004, that is, after the new collaboration had been established:

"The quality improved radically as the process moved ahead."
(Jesper, Test Expert)

After the component testing and the mini-flow testing, application, integration and acceptance testing should be performed in order. However, previous problems in the project and organizational difficulties regarding different tests had consumed more time than anticipated and resulted in that there was too little time left to run the different tests sequentially; the testers had no choice but to run them in parallel. This is described in the following section.

6.2 Forced to break the waterfall method

Although the plan was to perform the different tests sequentially, lack of time resulted in that various tests had to be conducted simultaneously. Since the testing was the last step in the chain it was always in the end the testers' time that was shortened, the testers argued, and therefore there was no alternative than to perform all tests simultaneously. In addition, the technical test environment and the hardware technicians was a bottle neck in the process. Too few people worked on installing test environments, updating and correcting test environments and test data. The technical infrastructure was complicated and resource demanding. Parallel testing was not a problem for application testers since application testing was first in the process (after component testing) but for those who performed integration tests it implied that defects turned up that should have been detected in the application tests, which resulted in that the integration tests were interrupted time after time. One tester said that it would have been better if the sequential order could have been kept. The acute remedy to handle this was careful coordination and frequent communication. The test group started each day with a morning meeting to inform and discuss the preceding day's errors as well as the coming day's tests and coordination challenges.

The same problem occurred when integration and acceptance tests were run in parallel. Acceptance testing is similar to integration testing in that both test whole flows of information and take an overall perspective on the systems. However, integration testing is technical oriented whereas acceptance testing takes its starting point in real operational situations; it investigates the system from the pension processes and customer service's perspective. Acceptance testing was supposed to be conducted after integration tests had been finished. Bugs were detected in the acceptance tests that should have been found already in the application testing, that is, in the test stage before the integration testing, several months ago. Especially the K2 system contained numerous bugs. This indicated that the whole project was very much behind the schedule. This kind of bugs should at this point in time and at this stage in the process not exist. And if the acceptance testers found so many application errors at one of the lowest levels of tests, how many bugs were still not detected? The Head Project Manager and many project members got worried since project time was running out.

The acceptance test leader realized that the acceptance tests could not be performed as planned. The solution in this situation was to reorganize the whole test procedure. The acceptance test leader paired some of the acceptance testers who had their background in the Operations Subproject (not in the "Test Project or Test Unit") with the application testers. Instead of testing only in accordance with the prepared and scripted acceptance test cases it was decided to "look around more widely" to explore the system with a broader view. Normally, the testers wrote test cases based on the requirement and design specifications to ensure that the right things were being tested and that the tests covered everything that should be tested. The testers still wrote test cases to ensure that the whole system was covered but while such tests were performed the testers also kept their eyes open for *unexpected* errors, unanticipated inconsistencies and unforeseen defects. In this way, different tests cases emerged and were developed more spontaneously as a consequence of prior tests and findings. For example, one test case regarded the "search functionality". A user should be able to search for pension savers in different ways, for example by name. When controlling this functionality the tester entered an "A" in the surname field. Many "Anderssons" turned up. However, what the tester also saw when "looking around more widely" was that those of the "Anderssons" that recently had changed address also were mistakenly registered as dead.

When investigating this further the testers found that the death date was the same as the day they moved. This bug would not have been detected at this point in time if they only had looked narrowly on the result of the test case "search pension savers by surname". In this way errors were detected quicker and the system was improved earlier than if the testers had continued in accordance with the scripts and the test plan.

Beside the time aspect, another reason to break the standard test procedure was that it turned up that some of the requirement and design specifications out of which the testers wrote the test cases had become old. The testers complained that the waterfall method with its bureaucracy made the information on requirement changes slow, which resulted in that the testers often created ambitious test cases and conducted time-consuming tests against old requirements. In this way much of the testers' time and effort was wasted. The testers had thus not received information in time about the changes. Since the acceptance testers, that is, the Operations Subproject members who were working in pairs with application testers were better informed and updated on late changes in the requirements they could quickly update the testers on these changes. The formal way of transmitting changes through the document bureaucracy took time which the project members did not have. An information shortcut through the waterfall process was created with the new collaboration form

Moreover, the normal pre-scripted testing based on the requirements and the design documents does not reveal any fundamental errors in the requirements or in the system architecture; it just checks whether the requirement specifications and the system solution match up, so this peer-collaboration and exploratory-like testing was also valuable to find deep errors and mistakes in the thinking behind the requirements and the system construction. The acceptance testers who came from the Operations Subproject were experts on the requirements and the pension processes and on how the system would be used in the future. They could detect fundamental errors in the requirements since they knew the processes and routines behind the requirements. The application testers on the other hand were experts in formulating situations where bugs in the code could be found and how things were interrelated and influenced each other and could quickly and easily write new clever test cases when needed. The cross-functional collaboration between people from the Operations Subproject who had been involved in the requirement process and the testers who knew how to intelligently check the system was fruitful in this time-pressured situation. One of the Operations Subproject Members said:

> "What has been tested earlier, what influences what and how I should perform the tests in different circumstances was hard for me to know. On the other hand, I knew what the end result should look like and how the users would use the system. I know how it was supposed to work and the reasoning behind the requirement specifications and therefore I knew 'this is wrong, this icon means this and not that'. I was inexperienced in designing test cases but learned during the way and Lena who helped me with the test cases did not know very much about how the telephony operations would work in reality Testers do not normally see the whole picture, how the things they test will be used in practice". Together we made up a good team.
>
> (Urban, Operations Subproject Member, Acceptance Tester)

A third positive consequence of the new test procedure was that the acceptance testers ("Operations project" members) interacted more with developers. Previously, the testers and developers had mostly communicated via a formal error reporting system, which had resulted in many misunderstandings, interpretation differences and interruption in the communication, and slow correction and retesting of bugs. Now because of lack of time the Operations Subproject members who assisted the test group initiated a more spontaneous and direct communication with the developers. Before a formal error report was written the testers talked to the actual developer and explained, demonstrated and gave a short introduction behind the test strategy and the goal of the test to speed up the process. The developer could then more quickly and easier go back and examine the problem in detail.

In sum, one can see that breaking the waterfall work model resulted in several positive consequences even though various testers said that their work situation became stressful and sometimes frustrating and confusing. The testers' time in the plan had been shortened for reasons that mostly depended on troubles in the project in earlier stages and because of this the testers had been forced to adapt quickly to the situation and abandon the normal test procedure. Both testers and developers and acceptance testers (Operations Subproject members) considered however that the intensified face-to-face collaboration was valuable and might be something to further develop in the future. Now, it remained to set the final pieces and end this project, which to a great extent involved the challenge to agree on how to deal with the bugs that the testers found in the different tests. This is explained in the next section.

6.3 Completing the work

One big recurring discussion was still alive – the one that concerned bugs and the bugs' relative importance. Different people had different perspectives and opinions on the errors and what should be corrected and what should be postponed. Section 6.3.1 starts with an explanation of the advisory board meetings, which were set up to formally deal and discuss and collectively decide how to do with faults to finally settle the work. Perspectives of bugs' importance differed quite often and it became important to find a collective reasoning that could unify different people's perspectives or serve as a guideline in the decision process on what should be corrected and how. The converging criteria that emerged in conversations at the advisory board meetings are described in 6.3.2. Then in the last section of the chapter, 6.3.3, the project members' experience from the implementation of the new system is described.

6.3.1 Diverging perspectives on bugs' importance and definition

From the summer 2004 when most pieces had been developed the discussions mostly regarded the bugs that were detected. Since time was scarce there was a rule that only the most important errors should be corrected and the rest of them postponed to a later version of the software system. In "theory" it sounds simple but in practice it was complicated since different people had different opinions on what was important and not. Discussions on what was right and wrong, if a problem actually was a problem, and who it was a problem for and the root of the problem and how important the bugs were, and which bugs should be corrected, interdependencies, and consequences were common. Often the error could be important from a user perspective while the developers at the same time could argue that it was just a minor fault. An error report system was used where the errors were described, classified and rated. However, many of the errors had to be analyzed and discussed further. For this purpose the project management had set up a weekly "Change Advisory Board" meeting. The project management and various experts convened to formally discuss and make decisions on current “open” errors. At these occasions the responsible developer also was expected to present potential consequences of correcting different bugs in terms of interdependencies and time required. During the project’s last months when all functionality was developed and the developers were mostly busy correcting and “closing” faults this type of meeting, so called Change Advisory Board meetings, was held everyday.

How did the project members do then to agree and make decisions regarding bug correction? Commonly, the project members evaluated potential solutions against available time in the project plan and discussed technical interdependencies and compared it to the consequences of maintaining the implemented solution. Sometimes the time factor was not enough as selection and decision criteria. One problem was that the time aspect did not take the user-friendliness perspective into account. After all, the project would not be considered successful if the users did not like it. As early as in the requirement phase it had been stated that the users’ point of view and satisfaction were of great significance for the success of the project. This belief turned up as a guideline in the choice between different solutions where time scarcity and technical complexity not alone aligned different people’s opinion. User significance and how frequent a certain feature or functionality would be used determined the importance of the bugs, together with the time aspect and technical complexity of a specific solution. This is explained in the following section.

6.3.2 Converging decision criteria emerging in conversation

In many situations the *user significance* determined which way to go. For example one member said; if a certain operation or “click” in the K2- system felt like it was taking long time to process, it also took too long time. The user experience was important to consider in different decisions to ensure that the new system would be well-received and easy to implement in the organization. People in the organization must think that this is a better system than the one they had before. This did not mean that the project changed everything in the way the users wished. In the first place the problem was discussed and the users were informed on what the changes would involve in terms of developing time and cost and asked how important the correction was in order to estimate the practical value of the problem and its correction. The users were asked if they could tolerate certain things and live with them. Features of the system that would be used *frequently* must function well in the users’ eyes; such things were important even though the problem was “minor” in the developers’ view, for example, too many “clicks” to reach a frequently used window or function could be annoying in a stressful situation. Or if a pop-up window places itself behind an activated open window so that the operator does not see it and thinks that he or she must click again to open it, might also be confusing and irritating. In the end the operator might have a dozen pop-up windows behind the activated surface without even knowing it. While frequency and significance errors were corrected, bigger problems related to unusual situations could be postponed for future developments.

The two factors explained above together with the time parameter influenced decisions and actions so often so that these factors in a way started to function as rules or praxis in the error discussions and decisions. I identified these factors in the conversations that took place during the change advisory board meetings. The conversation extract below concerns a problem with a reply-bottom that places the reply window behind the current activated surface. In this meeting the Quality Controller (QC) who seriously controlled that the project directives and the time parameter were followed, the Programming Leader responsible for the K2 system

(K2PL) and the overall Design Leader (DL), and the Head Project Manager (HPM) who contributed to the problem analysis and made decisions and ensured that the development process never stood still, and the Requirement Coordinator (RC) who was an expert on all pension processes and also well-acquainted in technical and information systems, participated. Marked sentences point at the decision criteria "user significance" and "user frequency".

The Quality Controller starts the discussion by reading the error report document and asks how important the problem is, in line 1 and 3. The developers do not think that the problem is so important that it needs to be corrected, in lines 4, 20 and say it is difficult to correct in line 10, 11. The Quality Controller understands that it is a problem from the users' perspective but realizes and reminds that time is very short, in line 16. The developers should deliver all corrections tomorrow. The error takes time and is complicated to correct and the Head Project Manager, The Quality Controller and the developers want to postpone it to a later system version, another project, lines 9, 13 and 14. However, the Requirement Coordinator argues that it is a big problem by referring to users' experience of the problem, in lines 6, 8, 15, 21, 23. It is tardy to click twice and confusing that the window does not show up. Only one click should be needed. The Requirement Coordinator says also that this function is used frequently, lines 17, 31, and 33. These constitute the reasons why the bug should be corrected. The discussion continues because the Requirement Coordinator strongly emphasizes how important it is for the users and how frequently the administrators will use the function, even if the other participants were ready to move on to the next error report. It turns out that if the problem is solved in a different way and so "reframed" and if another function of the same feature is de-prioritized, namely that you could choose language which is something that rarely happens in an operator's daily work, the problem becomes easier to fix, lines 25, 29, 30, 31, 44, 45. The two aspects, *user significance and user frequency*, that is how important it is for the users and how often the function is used played a crucial role in the decision whether the error should be corrected or not. The criteria contributed also to further analysis and helped the project members make a joint decision in a situation where the time parameter was insufficient but still important as steering parameter and decision criteria, lines 19, 38, 40, 41, 42, 45 and 46.

User significance and user frequency as converging decision criteria

1. QC: ”Write e-mail”- window pops up behind the administration surface. Do you know what this is?
2. K2PL: Yes
3. QC: But does it make a difference? Of course it makes a difference but how serious is this?
4. K2PL: I do not know, it only happens when pressing the answering button
5. HPM: Is it not possible to administrate or does one have to dig around or…?
6. **RC: One does not understand that it has been opened.**
7. K2PL: Oh yes, it is first opened and then it places itself at the back. You can see it first.
8. **RC: Yeah but I have worked with the administrators, I know that they do not understand that it has been opened because it took a while before we saw that there were 17 ”answer e-mail” windows in the lower part of the surface.**
9. HPM: Can we tolerate this and just make a clarification in…
10. RC: Is it difficult to correct?
11. K2PL: Yes
12. RC: I see. But then we do not have any choice?
13. QC: No, then it is a “rest”.
14. HPM: Can we make a clarification in the documentation that this can happen...?
15. **RC: It will always be an extra click which is tiresome.**
16. QC: Yes, I understand that this is not good but I feel that we must basically get everything corrected to the delivery tomorrow.
17. **RC: Yes, but this is a rather common thing that the administrators do. I am not very happy about this.**
18. K2PL: I can investigate how much work it is. I have already looked at this but I can check it again.
19. QC: Yes but...We don’t have much time to make a decision..
20. DL: But this is a blemish, RC.
21. **RC: No, it is not. It is really annoying for the administrators.**
22. DL: Yes yes but...
23. **RC: Listen, you click “answer” and then it falls down there and then one must click again.**
24. DL: But it can’t turn up behind, if it does so it must be because you have clicked in K2?
25. K2PL: No, the problem is that when you push the answer button...It is a special construction because it is a button with drop-down inbuilt.
26. DL: I see.
27. K2PL: And then it takes the focus so…When you have created..
28. DL: Yes
29. RC: But what kind of drop-down is it?
30. K2PL: If one wants to answer in Swedish or English. When you press the answer button you choose to...
31. **RC: Take the drop-down away, so it is only Swedish. How often does one answer in English, once in a… I mean, as an alternative solution, because it is Swedish e-mails up to 99% and then for this last percentage....**
32. K2PL: I thought it was common to answer in English; otherwise they would have called in…Those who send e-mails…
33. **RC: Maybe. But anyway, in relation to plausibility and how common this is…**
34. K2PL: But that was why...
35. HPM: Is a “survey the buttons” possible instead of…
36. K2PL: It will be two clicks anyway; first choosing the first and then the second.
37. RC: Or two buttons instead of one.
38. K2PL: Then we have to create new code. No...
39. HPM: No.
40. QC: I just feel that we have a pretty long list and we talk about delivery tomorrow.
41. HPM: Either we solve it today or we rest it.
42. QC: But if we solve it today we take time from something else that should have been solved today.
43. K2PL: But put it on “implementation”, I will take a look at it.
44. DL: Can’t you take RCs suggestion into consideration and see if you can take it away and use an ordinary bottom instead, if that is easier?
45. K2PL: Yes. Then it won’t be a problem.
46. DL: No because it is that one that takes quite some time.

By evaluating *user significance and user frequency* in relation to the error and change request, the management could reach a conclusion what to do and continue the process. These evaluation factors created some kind of general rules that repeatedly guided decision and collective action regarding errors and changes. There were no specific guidelines written in the project's development handbook, no detailed "codified law" how to deal with changes and errors, instead they had to create these temporary quick solutions or rules to handle the problems. These loose and unwritten "common laws" became precedential rules valid for several occasions but the rules were not as strict or binding as the things written in the project method handbook. The rules complemented the other evaluation factors; time, cost, and complexity. Thus people integrated their knowledge and created a new way of dealing with errors by "common law" rather than "codified law" when the project directives, development handbook, the steering parameter time or the priority list did not alone help them move forward to close and finish the errors and the whole development process. The project members did not explicitly refer to or talk about these judgment rules per se; the rules just implicitly evolved during the change advisory board meetings. Before the restart of the project errors turned up and changes were made both in the code and in the requirements over and over again, which made it difficult to go through the whole development process and reach the goal and end the project. It seemed that the communication in the change advisory board meetings was important to stitch the pieces of the system together and finish the development process and the project. The converging criteria that emerged during the discussion became thus a tool to finally close the process.

6.3.3 Successful implementation of a 'thousand pieces jigsaw puzzle'

As more and more things were finished in this "thousands pieces jigsaw puzzle" it became less difficult to coordinate activities and communicate and collaborate, one project member said. As the pieces fell into place and formed a picture it became easier to see how yet another piece could fit in. The project management had tried to separate the project from the ongoing businesses in the line organization during the process to be as little influenced by the activities in the line organization as possible. However, upon reaching the implementation day the project started to interact with the line in order to prepare the host organization to take over the system. The preparatory interaction consisted of different activities such as documentation, security controls, maintenance and service management, performance, robustness, and stress testing. One important part was also the users' education and training. Training end users to navigate the system, presenting the system to line managers in demo seminars and involving future application managers in the work several months before implementation resulted in that the members of the host organization were well accustomed with the new system when it was time to implement it. The system had been legitimized in the organization. In addition, through the early education, training and stress testing the project management received multi-dimensional feedback (social, technical, and human-computer interaction) before the system was put into practice, which could be met and treated before implementation to steer people's expectations. Human-computer interactive difficulties and mental and social resistance towards the system were dealt with which resulted in organization-wide acceptance and smooth implementation. Technically, the launch was successful and only few and minor errors were detected after the system had been implemented.

CHAPTER V

V. INTERPRETATIVE ABRIDGEMENT

1. Introduction

This chapter highlights the preceding chapter's most important parts from a knowledge integration perspective. It focuses on crucial organizational problems and the imaginative solutions that the project members have invented to solve such problems. The most critical challenges and solutions in each leg are described individually. The problems and challenges were selected partly from the point of view of the project members – what they argued for were the most difficult and important problems in their daily work and partly from a theoretical standpoint – what seemed relevant from a knowledge integration perspective. This chapter is thus a summarized interpretation of the case.

2. Leg 1

The first section includes the project's initial period (from autumn 2002 to spring 2003), the restart of the project, and "leg 1", which comprised the first step in the waterfall work model, which included the work with the requirement specifications. The first section contains a discussion of two problematic situations and how the participants worked their way out of these situations. In section 2.1, I discuss the problem of the organizational disorder that prevailed in 2002 and 2003 which was solved with the restart or "rebirth". In section 2.2, I describe the problem of the participants' limited comprehension of the task and its solution. The solution consisted of clarification of the task and the requirement specifications by integrating diverse people's knowledge into a simultaneous process..

2.1 Rebirth

The first problem that the project ran into was the organizational disorder that existed in 2002 and early 2003.The developers and project members who specified the requirements repeatedly disagreed and misunderstood each other regarding what should be done, when, and how. The participants also had problems with coordination and communication within the different groups. Moreover, the project goal had not been fully clarified and analysed, which resulted in that nobody had an overall view or understanding of the task. As a consequence, additional requirements were constantly specified and more and more system functionality was developed, at the same time as old versions were changed and updated. This nonstop development did not seem to lead anywhere.

The way out of this situation was to save the general idea and mission but disband the old organization and form a new one. The Requirement Project and the IT-project were merged into a single new project organization which was to apply a waterfall-like development method. During this "rebirth", new managers were employed. The project members called this the "restart" of the project. The computer programming was stopped since the first step in the new model was to specify and finish the requirements and so clarify the task and the goal. The requirements were, in turn, translated into a comprehensive project plan with an end date set, milestones, and different work packages and activities were specified in terms of time and earmarked resources. This was an attempt to better define the problem and clarify it before attempting to solve it. Instead of developing single requirements and pieces of system solutions one at the time, the project management now wanted to create a more complete understanding of the entire project. In addition, before programming the computer system, a construction drawing or system design was made to show how the participants intended to solve the problem and to make sure that the solution would match the requirements. In this way, one may say that the project members developed theories or hypotheses on paper about how the system would work before it was actually constructed, which, in turn, can be seen as a theoretical trial-and-error method.

Elaborating on the solutions in theory meant that different alternatives were now *not* tried out in practice. Bad alternatives (both in terms of requirement specifications and system solutions) and solutions that would be difficult to integrate were excluded earlier in the process, when compared to when specifications and features were directly programmed. The programming was not continued before the goal and the project plan had been established and convincingly presented in formal documents. The Head Project Manager started to protect the project from further changes; he decided to down-scale the organization and reduced the scope of the task and responded with a categorical “no” to every new external requirement and proposal, in order to be able to work in a sequential linear way and finish the project on time. Implemented together, these measures reduced complexity, uncertainty, and instability and enabled a faster and more simplified process. All in all, the way out of the organizational disorder was thus to restart the project - instead of continuing on the chosen road, the project members went back to start all over from the beginning again.

2.2 Open up for diversity to create a collective hypothesis

One part of the work with clarifying the task and improving understanding was to identify organizational processes, routines, and requirement specifications all at once, and to relate them to each other and to ensure the project’s documentation cohered at all levels. The task was defined, and structured at three different, but interrelated levels. These levels were the process-, routine-, and requirement levels. This generated a comprehensive picture of the *overall* pension system and the need for data system support, whilst, at the same time, as the *details* of each process and routine and specification were identified and analysed. This work required simultaneous integration of a wide range of skills and knowledge: laws and regulations, IT, pension administration, fund management, system user-friendliness, and communication. The user-perspective was particularly important. There was commonly a “real person”, for example, a pension-saver or an operator at the core of the discussion when a process, routine, or a requirement was developed. The project members’ imaginary thinking commonly included a pension-saver or support staff in a specific situation and then this person was followed through the different process stages or in different situations. This real-life perspective made it possible to grasp the whole from a process perspective, whilst different details could be analysed and elaborated on from various perspectives.

One may say that the problem of insufficiently understanding the task was solved by opening up the process to more diversity, for the synchronized integration of different perspectives. Diverse people from different departments convened in workshops to develop the processes and routines and requirements together. This work involved extracting thoughts, expanding ideas, and imagining situations *collectively*. The participants related their joint wealth of experience and knowledge to one other and envisaged how processes and routines could be materialized in a computer system. The coordinator juxtaposed people’s comments to generate a complete picture. In this form of meeting, the participants utilized each other’s knowledge and maintained their diverse perspectives. They engaged in a creative process together, without slipping into the same state of mind – the project members continued to analyze different situations from their diverse perspectives.

In order to enable the simultaneous participation of many different people from various departments more careful planning and preparation of meetings was needed. An agenda and goals were set for each meeting and workshop. The first period of the project which was characterised by an informal way of working, spontaneous communication, and ad-hoc problem-solving strategies often resulted in difficulties for people from different parts of the organization to prepare and meet at the same time. Confusion and misunderstandings were common. The planning gave them a chance to prepare and read relevant documents and think through the issue prior to the meeting. Having a goal with each meeting around which the project members could rally was also important, since the participants in this process did not have clear roles. The planning and preparation of workshops were also important, since the participants' knowledge-levels varied. It allowed those less experienced to learn about the demands set by the Requirement Coordinator and the other experts.

While the workshops were announced and planned in advance, there was still some informality to the way in which these meetings were run, in terms of conversational turn-taking, when and how questions were asked, and how statements were made. However, the communication and work procedures in the workshops were led by a facilitator (the Requirement Coordinator) so as to facilitate the exploration of details without losing the sense of the whole. The Requirement Coordinator kept things together and details were integrated in a specific conversation pattern, (which I called "integrative conversation") during the workshop sessions. The Requirement Coordinator synthesized and summarized the discussion every once in a while and related and integrated pieces to the whole picture. This drove the process forward. The Requirement Coordinator ended discussions with summary statements and decided when to move on and discuss new issues. The Requirement Coordinator and the "integrative conversation pattern" thus kept individual creativity and contributions under control.

Another activity that helped in solving the problem of insufficient understanding of the task and lack of overall view was peer-reviewing of the specifications. The peer-reviewing process also had a disciplining effect on the communication regarding the requirements. The requirement specifications were mostly written by the coordinator, except for a few parts. The other participants (especially future users, but no top managers) were asked to scrutinize and comment on the requirements. The documents should be written in a formal standardized way and electronically filed in the organization's archive. The documents steered the agenda and communication order during the review meetings. The coordinator collected comments and added and changed the requirements in accordance with the comments. Peer-reviewing became a natural way of working, and seemed to encourage a collective process of "help giving" and "help seeking" behaviour which resulted in a more elaborate requirement picture, when compared to the period before the restart, when thoughts and suggestions were not shared in a structured or formalised way. According to the project participants, The review process was crucial for the understanding and framing of the task and the legitimization of the system within the wider organization.

The requirement documents were delivered to the designers when the requirement members agreed and reached consensus around the specifications. Communication was channelled between the requirement people and designers through the Requirement Coordinator. This reduced the potential of individual requirement members to make changes. This made the Requirement Coordinator role even more important. The designers began translating the requirements into technical solutions and gave feedback on what was possible and economically sound to do. In this way, the designers continued with the peer-review process. Traditionally, in a waterfall method, each phase is closed before the next starts. That was the basic idea in this project, and was organized by the formal internal "handshake" between operations sub-project members before they handed over the requirements to the designers. The requirements however had to be altered until the design group accepted the requirements. The technical solution that the designers suggested had to be amended until the Operations Subproject members accepted it. Since the designers had an impact on the formulation of the requirements and likewise, because the Requirement Coordinator and the requirement members had a say on which technical solution was to be adopted, one may say that the waterfall method here was used in a "softer" way; the doors between the groups were left open. When both the requirement members and designers agreed on the requirements, as well as on the overall design of the system, the documents were formally "frozen" in what I call a "cross-boundary agreement" and could not further be changed by any of the parties. This made the situation less unstable, and the task and goal clearer. The cross-boundary agreement can be seen as the final and complete elucidation and setting of the task or "problem".

From a problem-solving and knowledge integration point of view, the process was about to convert practical insights and theoretical knowledge into a "new theory", or collectively accepted "preliminary conjecture" of how the new pension system and computer support system should function. As far as we know, at this stage no system was yet built. The project members had created a "hypothesis" on how the system would work. The hypothesis was not claimed to be correct or perfectly "true", but a well thought out, and collectively accepted, guess. Later in the process, this hypothesis would be developed and described logically in the software code, and then evaluated by the testers.

3. Leg 2

In leg 2, which comprised of the design and programming of the system two critical problems and solutions were identified. The first problem had to do with a lack of technical direction which caused technical confusion and integration problems. This problem and how the project management and members solved it by the creation of a "navigational aid" are described in 3.1. The second challenge that occurred in leg 2 concerned the organizational chaos and the collaboration clashes in the IT-groups. This was a result of a lack of leadership, the waterfall model, and the group dynamics and diversity among the developers. The way out of this difficult situation involved different measures that, together, overcame the turmoil and calmed the situation down and facilitated the project's progress. This is explained in 3.2.

3.1 Creating a navigational aid

Prior to the restart, there was no individual person who was responsible for the creation of the system architecture and system design drawings. The system solutions had often been directly programmed in direct response to the requirement documents. Even worse was that several things had been developed *without* requirements – these were technical solutions invented by the developers. The design became indirectly created by the implementation of the system solutions; it evolved from the implemented solutions almost as a consequence of the solutions. There was no cohesive structure or technical guidelines. This resulted in integration problems, confusion, and conflicts. Lack of technical direction and system drawings did not only cause problems internally among the developers but also with the Operations Subproject members in terms of misunderstandings, disputes, and instability in the system. The Operations Subproject members needed a response that the developers had understood what had been ordered, and a confirmation that these things would be delivered. Developers and Operations Subproject members, as well as managers, realized that the task or "problem" was more complex than anticipated and could not be solved until the interdependency of all its parts (the requirements and system parts) were understood.

The solution to this situation was the new design step in the waterfall model (the creation of overall and detailed design drawings). The design should show and prove on paper how the requirements were to be solved technically before anything was programmed and implemented. This development of the design can be described as a theoretical search for a complete integrated solution. The creation of the design documents aimed at detecting integration and construction problems and inconsistencies in solutions early in the process, before the system components were actually built. The development of the design was important because it made available the developers' work for collective examination. It seemed to function as a quality booster since "bad ideas" and incompatible solutions were sorted out early. The obligation to create a system design thus improved internal coordination among developers and the integration of different solutions.

The way forward became fixed in the design document and could not easily be changed after the "freezing" moment (described in Chapter 4). The creation of a design drawing involved a path choice, a "driving direction", or a map to which all new requirements and system solutions had to adjust. The design group and the design documents constituted a "chart room" – a place people needed to consult when developing technical routes or new functional trails. The architecture and design work became thus a way to build stability in the construction and was essential to the project's progress. The design also improved communication with the Operations Subproject members since it now became possible to show the Operations Subproject members how the system would be built. It was very important, according to the designers, to make sure that the Operations Subproject members and developers agreed with each other and had the same expectations on the system. In this way, the design work became an essential point of interaction. In summary, one may say that creating the design documents was like creating a "navigational aid" – a tool which provided the participants with a technical direction, and so helped in solving the problems that were experienced with the previous technical disorder. Creating a navigational aid helped the project members to coordinate actions and to find a common road towards the goal.

3.2 Balancing creativity and discipline

The next problematic situation that required different changes concerned the organizational chaos in the IT-project. The developers said that the technical leadership was weak and that communication did not work well between managers and developers. Furthermore, the new development work model was not being followed by all of the developers. Some of the developers continued to strive for perfection, developed system solutions on their own initiatives, and did not pay much attention to the detailed design task, the documentation, or the project's overall time priority. When the designers had finished their work and the design solution had been approved by the Operations Subproject members, the design drawings were communicated to the developers who were supposed to create a detailed design that was based on the high-level overall design documents. According to the new method, the developers should specify and document the details of the system solutions and then return the documents to the design group to get the details accepted, and, thereafter, build the system solutions. The overall design documents set the functional frames and limited the developers' alternatives. This sequential way of organizing the work almost killed the developers' creative possibilities and the resistance from the developers was immense. In addition, separating the overall design that was created by the designers and the detailed design, which should be created by the developers, also resulted in a "technical gap" and in a "time gap", as described in the preceding chapter.

The solution to this multi-faceted problem involved several changes, which mostly implied that developers' space for expressing diversity and creativity was limited. However, there were also two changes that actually allowed for more creativity and differences in opinions. This will be explained below.

The first action that limited creativity and diversity was to exchange leaders and members in the IT-project. A new Programming Leader was hired (the current Programming Leader left on parental leave) and one of the developers had to leave the team. The Head Project Manager took over the role as IT Subproject Manager. These changes in the workforce were supposed to decrease diversity in opinions and raise effectiveness in decision-making in the IT-subproject.

The interaction pattern among the developers changed with the installation of the new Programming Leader. In order to avoid whole group discussions the developers were to work individually or in pairs, and were assigned only limited tasks and responsibility. The new Programming Leader found out how to meet each developer, and argued, for instance, that one of them needed to be fed with new tasks constantly. The Programming Leader's reasoning here was that if a developer only had a minute left over, he would at once be lost in other creative work, trying to build some new "cool" technical solution. Important to note here was that the new Programming Leader ended discussions and made decisions, even if the developers disagreed and argued that the technical analysis had not been finished yet. The developers were not free to work under the principle that the best argument and solution would win if only open and honest face-to-face discussions were permitted. Open and free discussions did not seem to lead to progress, because every developer believed he had the best answer to every situation. Previously, discussions had often had no end.

To further improve decision effectiveness, a priority list was created by the project management, where the time was the most important steering parameter. It was declared that the job on hand was not about to find perfect solutions, but to create "good enough" ones, and that user functionality was prioritised. Non-functional and performance requirements were considered less important. This decision was made to coordinate actions and speed up the process. Developers could not afford the luxury of striving for the ultimate option, or lose themselves in a continuous search for better alternatives. Constantly looking for alternatives delayed the process and caused integration problems and confusion since solutions were interdependent and complex. If one thing were changed, often other things then also had to be changed. The developers' creativity and aspiration for perfection were thus also impeded by time scarcity, the priority list, and the "good enough" axiom.

Nevertheless, there were also some changes in the solution to the chaotic situation that allowed *more* creativity and diversity. The first measure was concerned with a change in the work that was done with the design documents. The designers and the developers decided to develop both the detailed design and overall design together. The developers' creativity and skills were better utilized in this way. There were no handovers between the design levels after this change in method. The steps of the development process were still the same; the design was created at two levels and both were documented, but the task was solved in a different manner. As a result of this change in the way of working, communication became more direct between the designers and the developers. They could exchange feedback in a more timely manner and ensure that actions were coordinated. This resulted in a reduction of the time gap. The new way of working also reduced the technical gap since both developers and designers examined the requirements. The developers got a better understanding of the end users' situation and could come up with better (user friendlier, more secure, cheaper) solutions. In total, the system's technical quality improved because the overall design and the detailed design now corresponded well to each other.

The second thing that seemed to foster creativity and the developers' diversity was the new meeting and communication style that evolved with the new managers and project members. Instead of conducting formal planning meetings, the regular status meeting time was used to discuss problems and different alternatives. The Head Project Manager and the new Programming Leader took a more active part in problem discussions and debates. The meetings hence changed from being status control and planning sessions, to being a supportive forum in which problems were discussed. In turn, this seemed to encourage the appearance of ideas, differences in opinions, and creative thinking.

3.2.1 Using emotional reasoning

Most important to this new form of communication that permitted creativity was the expression of emotions, metaphors, and analogies. Managers and leaders often had to make decisions and choose a path forward under uncertain and ambiguous conditions. In this project, it seemed that what the Programming Leaders and the Head Project Manager carefully listened to was, in addition to the project members' technical insights and well-informed opinions, the project members' affective judgment of a problem, situation, or solution alternative. The communication of an emotion from a project worker to the manager helped the manager appraise the direction of the project. Emotional expressions were used as signs of progress or regress. These emotional signals helped the managers to judge whether the project was heading in the right direction, or whether the search for another solution should continue. To watch out for these signs, by listening to different project members' rational as well as emotional evaluation over a period of time made it possible for the manager to react and act on project workers' inner knowledge. In the same vein, project members commonly used metaphors and analogies to improve understanding and facilitate discussion. The project members commonly created a metaphor or analogy of a problem to compare it with another more familiar problem to which all the participants in the conversation could relate. No matter how detailed, exact, or rational an explanation of a problem or viewpoint was, it did not always suffice to coordinate action. The IT-project members, therefore, often needed to go from a deep specialist knowledge level to another more general level, in order to meet and understand each other's opinions.

We can interpret this as the communication of tacit knowledge, because what the managers and participants were looking for was a hard-to-articulate intuition, a gut feeling, which is based on accumulated experience and knowledge. The "tapping" of emotions, or tacit knowledge, supported the project members' search for a good solution and also provided the leaders with accurate feedback on the status and progress of the project. If managers, for time and effectiveness reasons, cannot listen to all possibilities or learn in detail about each problem, they must "listen" to something else to get to know which way to go forward. In a "traditional and rational" world, intuition and emotion or passion-laden arguments and metaphorical speech would perhaps be discounted – in this project, emotional expressions concerning different problems and solutions were carefully taken into consideration, and even asked for. The Head Project Manager trusted these signals or “communicative markers”, and was frustrated when he did not get them. The emotional aspect of the conversations helped the project members understand each other’s problems, resolve troubles, select among different alternatives, and make decisions. When there is not enough room for perfect rational instrumental- and logical communication, or when this type of communication alone does not take the situation forward (because one does not know sufficiently about the things the other person refers to) it seems particularly valuable to try to complement the rational discussion with a more emotionally oriented communication. The metaphors and analogies and the emotional aspect of conversation, in terms of affective evaluation of the issue on hand, complemented the otherwise rational discussion. The conversation involved two types of exchanges which operated simultaneously; the rational exchange and the affective exchange.

In summary, the project management and the project members used several means to promote creativity in some parts of the work, while, at the same time applied other measures to limit creativity in some other parts of the work. One may say that diversity and creativity was not free, not completely restricted either. It was “bounded”.

4. Leg 3

In leg 3, the first problem of great importance concerned the inferior system quality and a resolution to break the “walls” between testers and developers by hiring a new test resource to work in close collaboration with one of the Programming Leaders. In this way, system weaknesses were overcome and the solution was substantiated early on in the process. This is described in 4.1. In 4.2, the second problem (and its solution) of this leg is explained. The second problem was related to the waterfall model, the shortage of time, and difficulties that were experienced in the test process. The way out of this situation was to abandon the sequential work model, and to conduct different tests simultaneously. One important ingredient in the solution was the change in collaboration, which involved more cross-functional work with improved possibilities to integrate diverse knowledge.

4.1 Introducing a test duet to substantiate up front

The first crucial problem situation that appeared in leg 2 (or more specifically between legs 2 and 3) was when the first program code was delivered to the test group. A conflict between testers and developers arose, concerning how the code was packaged, the missing delivery and installation instructions, and the quality of the code. This, in turn, led to a discussion of how much should be tested before delivery, and who should carry out the tests. There were differences of opinion between the testers and developers as to what the different tests should consist of, and there were differences in beliefs regarding the testers’ role and the test procedure. Seen from the project management’s perspective, the most important part of the problem was the inferior system quality, which the Head Project Manager focused on when he identified a way out of the situation.

The solution to this situation consisted of the invention of a new test concept which resolved the issue of what the different tests should involve. The solution to the problem also consisted of the recruitment of a new test resource who was to work closely together with the ELWIS Programming Leader to perform the kind of test that the project members had invented. The cross-functional “test duet” worked out very well and resulted in overcoming system weaknesses in program components. The testers became content with the quality in the code.

In addition, deliveries were improved in the sense that more formality was added to the process in terms of package, specifications, installations, and instructions. The deliveries also had to pass through the project office, the "tollgate", to be controlled and verified before reaching the test group. These measures, together, seemed to make the situation calmer and the collaboration among different project workers less contaminated with conflicts. Earlier deliveries of premature program features and functionality turned into well-tested deliveries of good quality. It resulted in an early and distinct quality enhancement; work that took a little more time and resources in the beginning but that paid off later on in the process. Coding and the early tests were now parts of the same step in the process, even though the developers still delivered within the group. The testers could thus, from now on, concentrate on running more complex tests where the entire application's features where examined. Fewer errors were inbuilt, which implied that testing, troubleshooting, corrections, and retesting activities were less interdependent, complicated, time-consuming, and costly than if the errors had been found later in the process.

The test duet generated skilful test cases, which can be seen as creating a fallible version of the system. The execution of the early tests and the critical inquiry of the system revealed the weaknesses of the solution. The developers resolved weaknesses immediately instead of waiting for formal test reports and change requests and new formal deliveries. The program code and the system were "substantiated up front". The logical reasoning expressed in the requirements, the design, the detailed design, and the program code were scrutinized and reinforced through this act of "upfront substantiation". The duet testing thus made the requirements with respect to how the pension system would work, as well as the problem solution – the technical system – more stable at this early phase of the project.

4.2 Introducing a cross-checking test approach

The second challenging situation that was encountered in leg 3 concerned the waterfall model and the time shortage. Some bugs in the program code had also not been detected when they should have been which signalled that the system had not been tested sufficiently at early test stages. This, in turn, meant that there was a risk that many other similar bugs still also existed. Overall, the test process was delayed for various reasons (lots of errors, test environment troubles, lack of hard ware equipment and technicians, slow testing, late changes/obsolete test data, and a slow and ineffective error reporting and correction system). The project managers argued that the sequential test process could no longer be used, since the system would then not be delivered on time.

To solve the problem, all different kinds of tests were conducted in parallel instead of sequentially. Various perspectives (user-, technical-, system integration-, and system independence perspectives) and approaches (pre-defined test cases and test scripts versus ad lib, spontaneous tests) were used simultaneously, which may be seen as a "cross-checking procedure". The system was examined and verified by using and comparing the result from different tests and sources. The most interesting part in the "cross-checking" procedure was the explorative test approach where the testers worked in pairs (the Operations Subproject members/Acceptance Testers were coupled with application testers from the Test Subproject). Many of the pre-determined test cases were abandoned in favour of a more open, improvisational approach where the testers looked around more widely. As on a discovery journey, just one test step was taken at a time, and the result from each step guided the next step. Consequently, there was more room for improvisation in this approach, than when using a pre-scripted test process. Different types of bugs were detected when compared to when isolated pieces of the program were tested, one category at a time, according to a scripted test procedure. This approach facilitated and accelerated bug detection. However, scripted and pre-planned tests were also conducted, and the combination of scripted testing and improvisational search was useful in ensuring wide test coverage and the detection of problems quickly.

The testers and the Operations Subproject members helped each other and complemented each other's knowledge. They learned and gained a more comprehensive understanding of the whole pension-IT-system process. The members from Operations Subproject learned what was possible to test, how to run tests, and different aspects that must be taken into consideration when designing a test case. Conversely, skilled technical testers from the test subproject got a deeper understanding of the operators' day-to-day businesses and what really mattered when different pension and fund transactions are dealt with and what was necessary to test carefully. When a tester knows how a function will be used in practice, he can, if it is necessary, create a shortcut through the tests and only test the most important and frequently used functions. If a tester does not know exactly how the system functionality will be used, then the tester must create comprehensive and complex tests which require a great deal of time. The "Operations Subproject Members" could thus through the pair collaboration help prioritize and contribute to a reduction in the total test time. The pair testing method speeded up the test process and the whole development process.

Another positive effect of putting together future users from the operations subproject with testers into pairs was that the end users knew about changes in requirements (that had been agreed on between the Requirement Coordinator and the developers) long before the changes were implemented as program functionality for the test group. This made it possible for the testers to redesign test cases earlier, than if the testers had to wait for change information and information reports. Before this change in the work method, the communication concerning requirement changes was very late and slow in coming, the testers argued. On several occasions, testers had been designing test cases and performing tests against old versions of design specs and requirement specifications unaware that changes had been made in the requirements. This was due to the fact that the document-based formal communication and delivery process was much slower than the fast working pace that the project members kept in reality.

The new form of collaboration also spurred on other team collaborations and cross-team communication. The deviation from test plans and test cases, the use of an integrative test approach, and the demands for more creative critical thinking put stronger pressure on more frequent communication among testers, and more informal collaboration between testers and developers. For example, in addition to discussing the execution of tests, what and how to design a test, what errors or problems existed, and what possible eventualities and scenarios within the pairs could emerge, all the testers collectively, across the teams, discussed and analyzed the bugs and the tests. A new standing meeting was organized where all the different testers participated and discussed reported errors. The new standing meeting resulted in the testers learning from each other and gaining more insight into details of different test approaches and ways of working.

Moreover, informal communication between testers and developers were intensified. The bugs were described, classified, and reported electronically and sent to the Programming Leaders, who then distributed the errors to the developers for correction. The system components were, thereafter, collected again and resent for testing in a patch delivery. By using this system, project members could see how many errors had been found, how many remained open, who had found the error, and where the error was in the correction process. However, the informal feedback from developers on what had been corrected and delivered to be retested was almost as important as the error discussion, one of the testers argued. Since there had been many problems with the test tools, environments, and deliveries, informal communication regarding the errors was crucial to analyze, trace, and locate the errors. (This included error in the code or in the test environment, or with the delivery and installation, and how the error occurred.). This helped the testers and developers organize and bring order to the test and correction work. The informal and quick communication on corrections also functioned as a constant progress feedback; participants were informed how much was completed and how much remained to be done in a quicker and more accurate manner than the correction system or the project plans could offer. In this stage of the project, it was more useful for a project manager to listen to the discussion regarding bugs correction if he or she wanted to know how the project proceeded, instead of looking at the plans or consulting the project office. Taken together, integration of different skills and more diversity and creativity in terms of cross-functional collaboration and informal and frequent communication were needed in this leg to deal with the work model problems.

Differences in meaning concerning the errors were still common and had to be solved, to end the project. Project management, testers, developers, different experts, and the Requirement Coordinator convened in the Change Advisory Board meetings to discuss important bugs and errors and to determine how to deal with them. There were different factors that influenced the decision; classification and priority, the existence of other errors, the time estimated to correct the error and the time available. When these aspects did not alone solve the problem, the importance of the error from a user perspective, in terms of “user significance” and “user frequency”, was used as additional guiding rules. This involved some pragmatic decision–making, where the end users’ perspective played an important formative role as decision criteria. Such criteria that limited diversity were essential to the completion of the project.

CHAPTER VI

VI. SYNTHESIS AND CONCEPT DEVELOPMENT

1. Synthesizing the project's problems and solutions

If we now put the observations made in legs 1, 2, and 3 in the preceding chapter together, we conclude that there were various problems during the process related to progress, collaboration, and diversity that the project members solved in different ways. The phases and activities of the development process that were used in this project (the sequential waterfall model) can, at an overall level, be likened to the stages in a *problem solving process*. As indicated in the table below, the work with the requirements can be seen as a problem formulation and identification activity (project members explore and define what system functionality and features are needed). The design and programming work can be compared to a solution search process (project members investigate how the system should work and build a solution in order to implement the requirements), whilst the test phase can be understood as a process where the solution is evaluated (project members examine whether the system solution solves the requirements, the needs or "problem"):

Requirement Specification	**Design and Programming**	**Test and evaluation**
Problem formulation	*Solution Search*	*Evaluation of solution*

Development process

Table 3: The development and problem solving process

The comparison with the stages of a problem solving process can facilitate the examination and understanding how different activities and tasks (specifying requirements, designing and programming and testing the system) were approached and how different "sub-problems" and solutions turned up and were resolved during the course of the project. The problem solving process comparison thus supports the structuring of the ISD process and the understanding of the different challenges that each phase involved. One must remember however that some problems and solutions turned up at the "boundary" between two phases or between two subproject groups, which made the process steps or phases less sequential and linear and the borders between groups less clear. Nevertheless, the problem-solving approach offers inspiration to the identification of the problem-solving and knowledge integration strategies and mechanisms that emerged during the process.

Below is a summarizing table with the problems and solutions and strategies that were invented and used in the different activities and stages of the project process. Each problem-solving strategy and solution, represented in the green-coloured columns as "way out", corresponded to a certain problem or problematic situation, stated in the red-coloured columns. The problems and the different solutions are briefly explained below.

Requirement Specification		**Design and Programming**		**Test and evaluation**	
Problem formulation		*Solution search*		*Evaluation of solution*	
Problem	**“Way out”**	**Problem**	**“Way out”**	**Problem**	**“Way out”**
1. No progress the first year	1. Restart the project	3. No direction	3. Create navigational aid	5. Inferior system quality	5. Substantiate up frontally
2. Inadequate problem representation	2. Open up for diversity	4. Detrimental conflicts	4. Bound creativity	6. Slow system (dis)confirmation	6. Cross-check the system

Development process

Table 4: Problems and “ways out” in the different development phases

During the first year of the development process, different problems emerged, as described previously, which resulted in the project not progressing. No useful system solutions were created (problem 1 in Table 4). In fact, the system requirements were poorly described and poorly understood, according to several project members. The “problem” to be solved was improperly represented, which is represented in the table as problem 2. The way out of this situation was to resume the project and start all over again (in the table “way out” 1). The project members who worked on the requirement specifications opened up the requirement process, and integrated more viewpoints and perspectives into the formulation of the requirement specifications (“way out” 2) in order to better comprehend the task at hand and reformulate the “question”.

One of the main problems regarding the design work and programming work during the first year of the development process was that the developers created individual solutions which did not match requirement specifications. These solutions were also difficult to align and integrate into a functioning whole. Neither did the developers have a general direction or general frame against which they could coordinate their actions (problem 3 in table 4). This also resulted in communication problems with the requirement team since the developers could not show how they intended to solve the problem as a whole. The solution to this situation was to form a design team consisting of three designers who were responsible for the creation and documentation of the system architecture and overall design (“way out” number 3). The developers no longer worked “silently” and “on-line”, but instead articulated their ideas and made them “visible” on paper. The programming solutions were accepted by the designers before the system was actually built. The designers created a “navigational aid” for the project, in order to form a direction and collective idea on which way to proceed. This aid became something to which the developers could add and integrate their contributions in a more coherent fashion than they did before, when everyone worked independently. In addition, the developers could better communicate with the requirement team how they planned to solve and implement the requirements.

The designers and programmers constituted a crew of diverse people with conflicting perspectives, priorities, and ideas on what and how things should be done. This resulted in negative conflicts (problem 4 in table 4) that resulted in stagnation in the development process. Several developers asked for stronger technical leadership to get more direction, end discussions, and improve results. The way out of this situation (“way out” 4) was to modulate the possibilities for expressing difference and individual creativity in certain respects. The project management limited creativity by creating measures by which the space for diversity and creativity could be regulated and adjusted in accordance with what the actual situation demanded. This is represented in the table as “way out” 4, “bound creativity”.

Concerning the “test and evaluation” development process stage, testers and developers experienced a cross-functional problem related to divergent work norms and values, which, in turn, resulted in an inferior system quality (problem 5 in table 4). This was solved by conducting system component tests early in the development process, before the development stage was closed and the system formally handed over to the test group (way out 6 in the table). The ELWIS Programming Leader and a technical test expert constituted a “test duet”, and collaborated closely to substantiate the system and raise the quality. In this way, the early type of testing became a part of the programming work.

Project members also ran into difficulties regarding the quality of the system and time limitations (problem 6 in table 4). Many different errors that should have been identified earlier in the process were detected late in the process, according to different project members. There was a concern that there could be even more and more serious faults in the program that then not yet were discovered. In addition, time was running out, and the project members realized that they could not continue testing by the book, that is, sequentially, at an increasing rate or level of system aggregation, as planned. In this situation, the testers decided to go “off track” and perform different kinds of tests in parallel, using a “cross-checking” strategy with different sources and methods. This is represented as “way out” 6 in the table, “cross-check”. By this measure, the project members managed to finish on time and delivered a useful system of satisfactory quality.

Altogether, the strategies, or “ways out”, corresponded to different problematic situations and evolved and varied over time in the project’s development process. These “ways out” will be more deeply interpreted, elaborated on, and conceptualised in the next section.

2. Collective Heuristics

This section develops an understanding of the solutions and strategies, the “ways out” that the project members invented and used to solve problems, integrate, and create knowledge and move forward through the different stages towards the completion of the project. Instead of calling the solutions, problem-solving strategies, or measures invented to solve problems as “ways out”, they will be identified and called “Collective Heuristics” (CH). This line of reasoning will be justified in the following section, 2.1. The table now shows that the “ways out” in the different stages of the development process (explained in the previous section) have been replaced by the concept “Collective Heuristics” (CH).

Requirement Specification		**Design and Programming**		**Test and evaluation**	
Problem formulation		*Solution Search*		*Evaluation of solution*	
Problem	**CH**	**Problem**	**CH**	**Problem**	**CH**

Development process

Table 5: The development process, problems and collective heuristics

This concept will be introduced in section 2.1 by featuring collective heuristics as problem-driven, emergent, and process-related. Then, in section 2.2, how collective heuristics facilitated knowledge integration by regulating the space for expressing diversity and creativity appropriately over time is discussed. In section 2.3, CH:s are examined as “time savers”. Saving time augmented the value of the collective heuristics for the project members, because the project was to be completed in a limited amount of time. In section 2.4, I summarize the chapter by presenting a number of remarks on the concept, “collective heuristics”.

2.1 Concept suggestion

Collective heuristics are to be understood as collaborative problem-solving strategies, mechanisms, or rules of thumb that the project members invented when they experienced problems that could not only be resolved by “ordinary” knowledge integration mechanisms, such as rules, directives, and routines, or development project means and tools, including project planning and reporting techniques, development models, documents or artefacts. Collective heuristics emerged and were valuable when project members ran into severe difficulties or reached an impasse and had difficulty in finding a way out using existing work frames and routines. The collective heuristics were problem-driven and became vehicles for project members to overcome obstacles, and allow the project to progress. Collective heuristics helped the project members to walk through the stages in the development process. For example, there was no point in entering the design and programming stage and carrying out those activities before the project members had “restarted” the project and “opened up the process” of specifying the project’s requirements. This involved the reformulation of the problem and the inclusion of more people’s skills and perspectives into the definition of the problem. Only once this had been done did it become meaningful to enter into the next stage of the process and suggest a solution to the entire problem. Likewise, when the project members had solved the problems in the design and programming phase, by using the collective heuristics “create a navigational aid” and “bound creativity”, and had completed the solution search process could they move on to the next phase and start to test and evaluate the system. Delivering disparate pieces of the system to the test group *before* the collaboration and integration problems had been solved was meaningless, because the quality of the system components was so bad and could not be integrated, which implied that the testers could not perform meaningful tests anyway.

By looking at the implementation of the collective heuristics over time, we can see that project members adapted the way they collaborated and organized the work as a consequence of what they experienced in each stage or activity. Different “norms” and principles were important in different situations. For example, when identifying and formulating the problem, project members generally considered that a restart was needed and requirement members decided to work mostly together in workshops and valued the simultaneous integration of different perspectives. Concerning the development of the system solution, developers cared for their creative freedom but also recognized the need for direction and joint action. Testers, in turn, appraised cross-functional collaboration, especially with end users, but also remarked on the significance of keeping a certain distance from the developers and the systems that they were to examine. Since the project members invented a collective heuristic to resolve a problematic situation, one might say that the project members learned by or “trial and error”. Thus the collective heuristics involved an experienced-based learning process. The different CH:s are featured as problem-driven process drivers that were invented to enable a faster movement through the development process. This is seen as the first essential quality of the concept. Moreover, the collective heuristics both promoted and restrained diversity in relation to what the actual situation required, and thus facilitated knowledge integration. This is the other essential characteristic of the concept, which is discussed in section 2.2.

2.2 CH pattern and diversity management

As we have now seen, the collective heuristics emerged and varied over time in the development process. Each one of them was a response to a certain problem. Additionally, the general pattern of collective heuristics also had an impact on diversity and unfolded as follows. In the first part of the process, when the task and problem was defined and requirement specifications identified, the project members “opened up” the process and involved different people from different departments. They widened the perspective that was eventually used to solve the problem of previously defining the problem too narrowly and in an unintegrated way. In this part of the project, when the system requirements were developed, the space for diversity was above all enlarged by the CH:s, thus enabling the participants to see the “whole picture”. In the table below, the left green column illustrates the collective heuristics that were invented during the requirement specification and problem formulation phase increased the space for expressing diversity and permitted more diversity. It concludes with “diversity mostly promoted” in the bottom row.

When the designers and programmers sought to find a solution to the problem, the CH pattern shows that the opportunities to express dissent were mostly restrained, and creativity was bounded by different means. The creation of a navigational aid, for example, the overall design and architecture documents, restricted programmers possibilities to develop own ideas, since their ideas had to fit in with the general architecture that the designers specified. In addition, the conflicts within the IT groups were managed by different formal changes and interventions; a CH called “bound creativity” (changes in team composition, stronger leadership, another communication format, priority list, and a strong focus on time and deadlines), that mostly decreased the room for divergence and disparity. In the table below, this is depicted in the red column whose bottom row is labelled “diversity mostly restricted”.

Requirement Specification		Design and Programming		Test and evaluation	
Problem formulation		*Solution search*		*Evaluation of solution*	
Problem	**CH**	**Problem**	**CH**	**Problem**	**CH**
No progress	Restart	No direction	Create navigational aid	Inferior system quality	Substantiate up frontally
Inadequate problem representation	Open up for diversity (Including "integrative conv." + "peer – review")	Detrimental conflicts	Bound creativity (Including "tap tacit knowledge")	Slow system (dis)confirmation	Cross-check (Including "use pragmatic evaluation criteria")
Diversity mostly promoted		**Diversity mostly restricted**		**Diversity mostly promoted**	

Development process

Table 6: CH pattern and Diversity Management

When the project members started to test the system, they found that the quality of the system solutions were inferior and needed to be improved. The Head Project Manager decided to recruit a new test expert to collaborate closely with one of the Programming Leaders, to "substantiate up front". This can be interpreted as an increase in diversity since the tester and the programmer knew different things and added their diverse skills and perspectives together to accomplish the task. In later test stages of the project, when problems related to both the test procedure and the system itself emerged, diverse people (different testers and end users) were also put together in order to approach and examine the system from various perspectives and with different methods. This is described as "cross-checking". This may also be understood as a way to increase diversity and apply diverse skills in the evaluation stage of the problem-solving process. Taken together, the CH pattern in this phase mostly involved an increase of diversity. This is showed in the green column to the right in the table above, "diversity mostly promoted".

The project members, especially the developers, constituted a group of diverse people, whose space for expressing creativity and diversity was limited *but also* in fact stimulated, although in a controlled manner, by the invention and use of different CH:s. Diversity may be increased and decreased in the same situation. It appeared that, where diversity was promoted, there were also disciplining mechanisms in use. In phases where diversity was decreased, creativity enabling mechanisms were also at work. For example, as is shown in table 6 above, when specifying requirements the diversity was mostly raised, as stated before, but diversity and creativity was not left uncontrolled; individuals' contributions were synchronized and coherently combined in specific integrative conversations that took place during the requirement workshops and seminars (as described in previous chapter). With respect to the design and programming work, diversity was mostly limited, in order to integrate the developers' contributions. However, since the task and the search for a useful solution required diverse expertise and creativity, the project management encouraged, listened, and tried to act heedfully to the different members' tacit knowledge that was often expressed in emotions, metaphors, and analogies. In test-related activities, diversity was, in general, increased. Although, in order to speed up the process and come to a project end, alternatives, trials, and discussions were closed down by the use of specific pragmatic evaluation criteria that emerged in practical reasoning conversation processes. In this way, imagination and discipline went hand-in-hand throughout the project process.

In the next section, 2.3, CH:s are discussed as time-savers, which can be seen as a third feature that was important to the project management and the project members since from the restart they experienced a strong time pressure.

2.3 CH:s as time-savers

The collective heuristics that were invented and used during the process did not only serve as collaborative problem-solving strategies, they also sped up the knowledge integration process in different problematic situations, which was important since the project operated under a severe time pressure. For example, the restart of the project and the changes that were made in the development process and the organizational design may have quickened the process (even if the restart implied that the process first was halted), since it resulted in the breaking of vicious circles and time-consuming iterations of requirements and system solutions. People from different departments were brought together in workshops, peer-review seminars, and cross-boundary meetings to combine their knowledge and apply their different skills simultaneously "in the same here and now" to create a wider and more beneficial formulation of the problem. In this stage, additional diversity accelerated problem-solving since it facilitated the creation of a more appropriate problem representation.

Requirement documents and design descriptions were accepted by both requirement members and IT members before the system was constructed and implemented. This resulted in the avoidance of late and time-consuming changes in requirements and code.

The new representation of the problem also made it easier for programmers and designers to create a coherent solution to the problem on paper, before programming the system. This resulted in a reduction of the risk of creating incompatible solutions, which saved the project time in the long run.

Furthermore, the restart entailed that only one steering parameter, *time*, was chosen instead of having three parameters which may have pointed the project members in different directions. This speeded up the process since the one steering parameter resulted in a clearer direction and coordination of actions. The focus on time also speeded up the process in itself since project members had to prioritise among requirements, delimit the scope of the project, and basically work longer days, more quickly, and in a more structured fashion (waterfall model, handovers, documentation, pre-planned workshops, project plans, regular status meetings and problem-solving meetings). This resulted in fewer changes in the project requirements and in the code.

The collective heuristics that were implemented to facilitate design and programming activities, "create a navigational aid" and "bound creativity", hastened the problem-solving process by decreasing diversity, aligning different individuals' contributions concurrently, and integrating the contributions into a functioning whole. This thus differed from the CH:s in the problem formulation stage which increased the pace by permitting and adding more diversity to the process. The CH:s in the solution search stage sped the progress up by decreasing diversity. The "navigational aid" pointed out the way ahead; a direction, to which the developers must adjust and form their contributions. "Bound creativity" signified that the project members implemented some changes (the team's composition including leaders, the design work, the way the developers communicated, and the priority list) that generally decreased the space for expressing diversity and made collaboration and knowledge integration easier. Although the changes in the design work permitted more creativity and diversity, they saved time anyway since the time lag that the sequential model had caused was reduced when the developers' contributions were integrated simultaneously.

Finally, as stated earlier, the CH:s "upfront substantiation" and "cross-checking" improved system quality and shortened the total test time. Different tests were carried out in parallel by different people who applied their different skills in close collaboration. More errors and different types of errors were detected earlier through this type of testing, according to the project members, if compared to what would have happened if they had continued with the sequential test approach. This saved the project on the whole some time.

In summary, the collective heuristics served, in a flexible way, to fine-tune and manage diversity in knowledge integration and problem-solving processes in this time-pressured complex development setting.
In the last section of this chapter, section 2.4, some reflections on the notion "collective heuristics" are discussed in terms of what a CH really is and what it is not, and how one can understand the parallel between the project, problem-solving and knowledge integration processes.

2.4 Summing up the Collective Heuristics – reflections

What is a CH? When a problem that hindered project progress emerged, an ad hoc, temporary, and broad work principle, a "collective heuristic" (CH), was invented. The concept "collective heuristic" is built upon its power to solve the various collaboration problems and knowledge integration problems that appeared when the project members were engaged in different project activities. A CH can thus be seen as a response to a problematic situation. The collective heuristics identified here were all related to collaboration problems and knowledge integration problems, as well as to technical aspects of the project task. The concept 'collective heuristic' therefore stands for the methods that the project members invented and used to integrate their diverse knowledge and solve collaboration problems. The word "collective" in collective heuristic was chosen because the solution to these problems involved collaborative processes rather than individual problem-solving methods. The word *heuristic* comes from the Greek word for "find" or "discover", and stands for the strategies or rules of thumb that individuals create and use to solve complex problems and make decisions in problematic situations.

The collective heuristics helped the project members to solve problems, to find new ways of working, and to integrate their diverse knowledge and move on through the process. CH:s facilitated the project's progression and can, therefore, be characterized as problem-oriented process drivers. Some of the CH:s, for example, the "restart" and the "upfront substantiation", were only useful once and were abandoned as soon as the problem was solved and the task finished. The CH:s might be described as flexible and dynamic knowledge integration enablers that changed with the prevailing situation and actual problem. The CH:s involved different knowledge integration processes or "practices", such as the workshops and the integrative conversations as efforts to open up for diversity. The CH:s were tightly connected with change and action in the sense that they accomplished transformations in the current way of working and integrating knowledge. However, the CH:s entailed some structuring elements as well. One example of this was the formation of new sub-groups, such as the design team and the cross-checking test pairs.

In general, the CH:s constituted satisfying solutions but they were not "analysed" or "calculated" before their implementation, which means that other solutions or strategies may have worked out even better. However, the time pressure made a more comprehensive search for an "optimal" solution difficult and the collective heuristics thus seemed to be "good enough" alternatives.

One specific feature and value of each collective heuristic was its impact on diversity and creativity. Collective heuristics appeared to help strike a balance over the project's life-time between promoting and utilizing diversity whilst, at the same time, progressing effectively.

The CH:s were identified in this one particular setting, but their application need not to be local in time and space; there may be some general power in each of the CH:s that were identified in this study that can be used in other development settings. However, the "dry land swimming" and "creation of a trial product" (described in Chapter 4) that can be seen as collective heuristics remind us that CH:s do not guarantee success. CH:s can even result in failures or mistakes. The CH:s should be best seen as general strategies that are worth trying out in time-pressured situations when one experiences difficulty with identifying the problem, searching for a solution, or evaluating a solution.

I identified and selected two crucial problems and two important collective heuristics within each problem-solving phase, but there could have been more or fewer problems and collective heuristics involved at the same time in each phase. In new product development and innovation, project managers and project members cannot always know beforehand what or when problems will turn up, and whether "old" proven solutions can be reused, or if new ad hoc local problem-specific strategies, such as collective heuristics, must be invented and implemented quickly.

As mentioned above, the CH:s were identified and associated with particular project activities and stages of the problem solving process. For instance, it would have been difficult for the project members to "cross-check" the solution *before* the problem had been accurately defined and represented. On the other hand, the collective heuristics "restart", "open up for diversity", "create a navigational aid", and "bound creativity" were identified and presented in relation to a specific phase but need not be strictly related to a particular phase in the problem-solving process. The different CH:s, as well as the activities in the process, were not as separated and linearly organized as the table in this chapter may suggest; they overlapped to some extent, and project members carried out some of the activities in parallel. Several CH:s were in use simultaneously. CH:s evolve in relation to difficult collaboration situations and are to be understood as problem-driven in the first place, which means that they do not have to appear in a specific order.

What is not a CH? In order to further specify the notion, I will shortly discuss what I do not see in this concept, that is to say, what does not feature a CH. First of all, this notion does not refer to individual problem-solving or decision strategies, as mentioned above. My take on the concept thus differs from the original theories (e.g. Herbert Simon and Daniel Kahneman) behind the concept "heuristics". Second, the collective heuristics identified in this study do not claim to offer optimal solutions or the "best" solutions to the different problems. Heuristic problem-solving does not guarantee success or the "right" answer, or even a beneficial solution at all, as described in the theoretical chapter. Third, collective heuristics do not function as routines or specific rules that have been developed to carry out a particular task in a certain way. The collective heuristics were more like rough "one shot" measures implemented to solve or "deal with" a specific problem. Collective heuristics are not complete strategies that prescribe in detail how a problem should be solved or how a task should be conducted. The CH:s were mostly broad and allowed the project members to work out the details on the fly. Fourth, the CH:s were, as claimed above, problem-driven and process-oriented and thus did not constitute a predefined structure that specified in advance how to organize and coordinate the work.

In the next chapter, the CH:s are presented as a complement to knowledge integration mechanisms. The relation between CH:s and KI mechanisms will be explained. I will also identify the different knowledge integration mechanisms used in the different phases of the project.

CHAPTER VII

VII. COLLECTIVE HEURISTICS AND KNOWLEDGE INTEGRATION

So far I have claimed that CH:s emerge when traditional knowledge integration mechanisms and project management tools are insufficient; when project members had problems that could not easily be overcome within existing project structures, methods, models, or tools. In this chapter, the relationship between CH:s and KI mechanisms will be further examined. This entails an identification of the knowledge integration mechanisms, activities, and agents that were applied in this project and an examination of how they changed over time. The chapter is structured as follows. First, in section 1, I present an overall figure of the whole project process, the problems and the CH:s and the KI mechanisms, processes, activities, and agents that were identified in each phase. The KI mechanisms, processes, activities, and agents are referred to as the 'Knowledge Integration Apparatus' (KIA). As we will see, the KI Apparatus varied over time along with the problems and the CH:s that were invented and implemented in the different phases of the project. In sections 2-4, I will discuss the details of each phase. Last, a short summary is presented in section 5.

1. Picturing the CH and KIA process

The main purpose of figure 11 below is to give the reader an overall picture of the development in the KIA over time. It shows the three main sequential steps in the development process; "Writing Requirements" (the green arrow up to the left), the "Design and Programming" (the red arrow below the green arrow) and Testing (the purple arrow down to the left). These steps correspond to the legs 1-3 described in the empirical chapter (Chapter 4). The three steps are further compared with the phases of a problem-solving process; *problem formulation*, *solution search*, and *evaluation of solution*, which were explained in the previous chapter. These steps are listed as subheadings to the three arrows in the model. Two critical problems and solutions have previously been identified in each phase. They are depicted to the right of each phase in the figure. The two CH:s in each stage are associated with a certain configuration of knowledge integration mechanisms, processes, and agents, a specific KIA, and are represented by the blue horizontal arrows in the figure. There were many different knowledge integration mechanisms, processes, and agents involved in each phase. The most important ones in each step label the blue arrows, the KIA, in each phase. In the Requirement Phase, the most significant knowledge integration mechanisms were 'sequencing' and 'group problem-solving'. In the Design and Programming Phase, the central elements of the KIA were 'certain agents' and 'directives', and in the Test Phase, the key integration mechanism consisted of group problem solving occurring in a mosaic fashion. There were actually many more knowledge integration mechanisms and activities in each stage, which are summarized in the bulleted list to the right of the figure. This will be described in detail in this chapter.

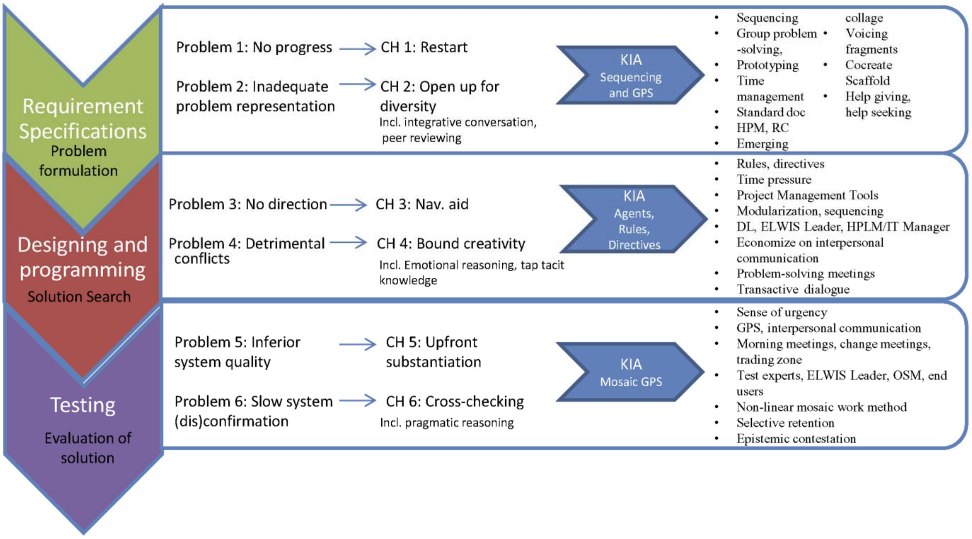

Requirement Specifications
Problem formulation
Designing and programming
Solution Search
Testing
Evaluation of solution
Problem 1: No progress
CH 1: Restart
Problem 2: Inadequate problem representation
CH 2: Open up for diversity
Incl. integrative conversation, peer reviewing
KIA
Sequencing and GPS
Sequencing
Group problem -solving,
Prototyping
Time management
Standard doc
HPM, RC
Emerging collage
Voicing fragments
Cocreate
Scaffold
Help giving, help seeking
Problem 3: No direction
CH 3: Nav. aid
Problem 4: Detrimental conflicts
CH 4: Bound creativity
Incl. Emotional reasoning, tap tacit knowledge
KIA
Agents, Rules, Directives
Rules, directives
Time pressure
Project Management Tools
Modularization, sequencing
DL, ELWIS Leader, HPLM/IT Manager
Economize on interpersonal communication
Problem-solving meetings
Transactive dialogue
Problem 5: Inferior system quality
CH 5: Upfront substantiation
Problem 6: Slow system (dis)confirmation
CH 6: Cross-checking
Incl. pragmatic reasoning
KIA
Mosaic GPS
Sense of urgency
GPS, interpersonal communication
Morning meetings, change meetings, trading zone
Test experts, ELWIS Leader, OSM, end users
Non-linear mosaic work method
Selective retention
Epistemic contestation

Figure 11: The CH:s and the KIA over time

Each section below is organized as follows. First, I provide a description of KI mechanisms in use. Then important KI agents and their contributions are explained, followed by an explanation of the different KI activities that were undertaken in each phase, and a description of the KI communication. Last, I present some reflections on how the knowledge integration process may be understood and conceptualised. In addition, one can find commentary on how the CH:s and the KIA effected the project members' possibilities to use their unique knowledge and express diverging opinions and perspectives.

2. The CH:s and the KIA in the Requirement Phase

During the course of the project, various CH:s were invented and used in order to solve the problems that the KI Apparatus alone could not resolve. CH:s thus complemented the existing KI Apparatus and sometimes also initiated changes in the KI apparatus. This will be explained in more detail along with an identification of the different sets of knowledge integration measures that were employed during the project.

During the project's first year, problem 1 emerged, "No progress", which was resolved by the CH "Restart" as explained previously. At this time, there was a general lack of knowledge integration mechanisms, agents, activities, and communication in the project team. For example, at an overall level, the development undertaking was organized into two functional projects instead of in one cross-functional project. The testers constituted a separate group belonging to a line function. Several project members reported that what existed at the time was a non-working iterative-like development process with unclear directives, rules and routines, disorganized group problem-solving, non-working formal communication, unclear roles, insufficient leadership, and unclear responsibility for integration of requirements and system solutions. Moreover, the Head Project Manager compared the developers and their uncoordinated work with a “bunch of dices that were just shaken and randomly thrown on the table”. In summary, there was a lack of mechanisms during the first year.

2.1 KI Mechanisms

The situation changed with the implementation of the CH “restart”. A new organizational design, a cross-functional project that comprised all three groups; requirement people, developers, and testers was established. The new project management formed and implemented a KI Apparatus consisting of a sequential development model, which can be seen as the KI mechanism “sequencing” (Grant 1996a). There were also cross-functional workshops and peer-review seminars, which can be interpreted as a form of organized “group problem-solving” (Grant 1996a), low-tech simulation and visualization, which can be compared to the “prototyping” KI mechanism (Schmickl and Kieser 2008), and a strong time focus, which can be seen as a the KI mechanism “managing time” (Okhuysen and Eisenhardt 2002). Moreover, standardization of different documents and media and the implementation of a more formal change management system were implemented and these can also be seen as KI mechanisms (Grant 1996a; Kellogg et al. 2006) since they coordinated action across the different subprojects (this is explained in 2.3).

2.2 KI Agents

The CH “restart” also gave birth to some new roles or “KI agents”. First, a new Head Project Manager (HPM) was recruited. He initiated and implemented the KI Apparatus and continued to lead the development process by managing the KI Apparatus, as well as diversity, creativity, and progress in different ways. Moreover, a new role called Requirement Coordinator (RC) was formed. The responsibility of this role was to specify, formulate, and integrate business requirements into a complete and coherent “requirement picture” and to make sure that coherent requirements were sent to the developers. The Requirement Coordinator was tasked to constitute the communication link between requirement members and designers/developers, and to ensure that the requirement members received a response from the designers in terms of an integrated software system proposal. Furthermore, the Requirement Coordinator played a crucial role as facilitator in the requirement workshops. She integrated her own ideas with other experts and future end users’ insight and knowledge, in order to create relevant business processes and system requirements.

The Head Project Manager and the Requirement Coordinator can be seen as “KI agents” similar to Berggren’s et al. (2011) KI agents in development settings, in the sense that these agents were knowledgeable and participated in all different parts of the development process and also understood how the system would be used in the future by the end users. The Head Project Manager and, especially, the Requirement Coordinator influenced how the requirements were formulated and translated into a computer system and , later on in the process, how the system was evaluated and tested. The Requirement Coordinator held a central position in the “cognitive network” (Kameda 1997) since she shared some knowledge with almost all of the other team members. She was able understand their perspectives and problems and, at the same time, express her own thoughts and ideas in a way that the different members understood. Because of this ability, she was considered to be an expert among the Requirement Project Members, in spite of the fact that she was not a specialist of a specific functional area. This observation is in agreement with Kameda’s et al. (1997) idea on that members with broad knowledge are commonly viewed as experts by other members, that is to say, “cognitive centrality” signifies expertise. Furthermore, her role and engagement as workshop leader and facilitator was self-appointed, but not questioned by the others. The Requirement Coordinator documented the requirements and thereby she greatly influenced the formulation of the requirements. According to Sawyer (2003), members who impose their own strong will in a problem-solving task will automatically reduce the other members' space for expressing thoughts and ideas. This may have happened here at times as some of the participants mentioned, especially since the project members operated under strong time pressure. Nevertheless, their general opinion was that their collaboration and joint performance was successful.

2.3 KI Activities and Communication

The second problem that emerged during the project's first year was identified as "Inadequate problem representation". The project members' remedy for this was the strategy or CH "Open up for more diversity" as described previously. This collective heuristic can be paralleled with the idea that diversified groups presumably generate a wider range of problem representations and solutions compared to groups whose members share the same background and skills (Dunbar 1997; Larson 2007). According to various researchers (e.g. Nickerson and Zenger 2004; Sheremata 2000; Tiwana and Mclean 2005; Iansiti 1995) the framing of complex problems often requires knowledge that is distributed among different specialists. Increasing and utilizing diversity involved, above all, one change in the KI activities that the requirement members carried out; a change from unstructured meetings to pre-planned and semi-structured requirement workshops. This is explained below.

Before the implementation of the CH:s "restart" and "open up for diversity", the communication among requirement members, and between requirement members and developers, was, according to the project members, rather unorganized and unplanned. Project members argued that this "open" and "free" interaction, which is often advocated when solving complex problems in development settings (e.g. Wheelwright & Clark 1992), caused confusion and made it difficult for them to act in concert and keep an overall view of the requirements and the system solutions. The requirement workshops and seminars were, therefore, more carefully planned in terms of a pre-set date, time, topic, and goal to be achieved. This also allowed for the participation of important experts. The pre-planned and semi-structured workshops thus constituted the arena where the process was "opened up for more diversity" and allowed for the simultaneous participation of a range of diverse experts from different departments. This increased the utilization of diversity. In his research, Stasser (1999) also found that structured meetings and discussions can reveal more unshared information and stimulate diversity in comparison with unstructured discussions.

In Kellogg's et al. (2006) terminology, the members made their perspectives and insights "visible" (using the whiteboard to explain different details of the business processes and routines), "accessible" (through simultaneous participation), and sufficiently "legible" (through verbal interpersonal communication) to co-workers. Together, they assembled and combined the various contributions into a growing "requirement picture", like an "emerging collage". This may also be likened to Majchrzak's et al. (2012) coordination practices; "voicing fragments" and "co-creating a scaffold". These practices implied that cross-functional members together created a common landscape out of their individual contributions and formed a preliminary representation of the product. During these more pre-planned structured workshops, interface problems that the project members could all see and discuss at the same time came to the fore. This is something which Schmickl and Kieser (2008) point out as being important for knowledge integration. Schmickl and Kieser (2008) emphasize the project members' need to discuss interface issues, but, at the same time, their need to reduce other interpersonal communication.

The participants of the workshops also used this arena to identify specific issues that needed to be more elaborated upon within their respective domain functions. Moreover, similar to the heuristic strategy "analogical reasoning" (Dunbar 1997), the Requirement Project Members sometimes solved problematic situations and issues by comparing the problem, or at least parts of it, to similar problems from the past. In this way, a number of transitional steps and alternatives were bypassed, and a solution was more quickly reached. This heuristic was used due to the fact that the members came from disparate backgrounds and brought different experiences together. The diverse group had thus broad material to work with and use in their problem-solving.

The CH "open up for diversity" involved two new KI activities that were to increase diversity and the integration of more perspectives; *the peer-review process* and *the elaboration of cross-boundary agreements*. The peer-review process increased the integration of the users' knowledge and perspectives, in specific. End user involvement was important since the system had to be accepted by its users. It had to be useful and purposeful and solve their every-day operations and problems, in order to be considered successful. We interpret this as if the project members had a goal-directed pragmatic view of the system that they developed (Cross & Sproull 2004; Cook & Brown 1999). The cross-boundary agreement entailed a stronger focus on the early inclusion of the designers and developers' knowledge. Peer-reviews and cross-boundary agreements were viewed as helpful control devices and can be compared to "help seeking" and "help giving" (Hargadon and Bechky 2006) activities. These activities accomplished a general "reformulation" of the problem and overall agreement on what the project task involved. Because of these changes, more diversity was permitted in this stage of the process.

The different processes and activities in the requirement stage of the project process began working out well when the project was "restarted" (CH 1) and the task was reformulated by the simultaneous integration of a wider range of perspectives (CH 2). In summary, one change in an existing KI mechanism (from unorganized and spontaneous group problem-solving to more structured meetings and communication) and two new KI activities (peer review and cross-boundary agreement) were implemented which, together, allowed for more diversity. However, whilst diversity was increased there were also KI elements inherent in these activities and knowledge integration mechanisms that controlled the range of diversity that was expressed. This will be explained in the following sub-section.

To make contributions from diverse people cohere, integrative and diversity control elements in the different mechanisms and activities were employed. In the workshops, an "integrative conversation pattern" evolved, which seemed to initially foster diversity and increase the chances for unique knowledge to be mentioned, since each issue was carefully and jointly investigated in a "transactive" manner (Teasley 1997); but then, in the following sequence, they were cautiously tied together. This pattern can be compared with Nonaka (1994) and Sawyer's (2003) ideas on conversational rhythms. Conversational rhythms first display a divergence phase with an examination of a problem from various perspectives, which then is followed by a convergent phase where aspects and occurrences are put together to form a coherent story. During the workshop sessions, project members hurried themselves, paid attention to deadlines and time available in relation to different tasks and occasionally they avoided raising more questions and ideas in order to move on quickly, as mentioned earlier.

Time pressure and work overload had thus to some extent a reducing impact on the opportunities to express diversity at the same time as it enabled faster integration of knowledge, which can be compared with Okhuysen and Eisenhardt (2002), Tyre et al. (2002); Gersick (1989) findings on time as a knowledge integration mechanism.

The peer-review process was centred on the requirement documents that had a standardized format. In general, they were written by the Requirement Coordinator, which decreased variation in the documents. The standardization of the requirement documents and the idea that the writing should be a one-person task, and that the same person constituted the sole communication link with the designers, can be likened to Kellogg's et al. (2006) research on standardization of media and communication forms and its reducing impact on variation, diversity, and creativity. In addition, the Requirement Coordinator role, the peer-review process and the standardization of documents implied a decrease in interpersonal communication and interaction. This can be compared to Enberg's et al. (2006) notion of the importance of economizing on interpersonal communication when diverse knowledge is to be integrated. The cross-boundary agreement took place in communication between the Requirement Coordinator and the designers, which was expected to reduce inconsistency in the documents compared to before the restart when anyone from the Requirement Project could communicate requirement specifications with the developers. Furthermore, diversity was also reduced in an interpersonal manner in the meetings, since the designers limited the requirement members' desire for new system features by arguing for what was possible and not. Similarly, the requirement members occasionally rejected the designers' system proposals when the suggestions included solutions that they considered did not fulfil the end users' needs. Finally, once an agreement was finished, no more changes could be made. Both sides had to adhere to the agreement, which also limited future space for diversity and creativity.

2.4 Reflections

The two CH:s, "restart" and "open up for diversity", showed that the project members had collectively decided and judged what relevant to include in the problem formulation. They engaged in a pre-selection process (Lindkvist et al. 2011). The problem formulation process also comprised of "variation" as well as "selective retention" (Lindkvist et al. 2011); variation in the sense that the project members did not know at the outset of the process what part of their knowledge would prove to be useful or in what way they could contribute. Notwithstanding this, they persevered until they reached a satisfying solution (for example, the requirement documents). Many project members only knew that they were to participate in this project, but not what their specific role would be. Neither did they know what was expected of them, or how they were to interact and integrate their diverse skills. The Requirement Project Members acted and contributed to the process in an improvised manner, as they found appropriate, within the frames of the more structured or organized workshop arena. A particular interaction pattern emerged and became more and more stable and tacitly understood over time. The requirement members argued that their experiences from the process were mainly positive after the "restart", and that they were surprised how much they had been able to contribute to the project and how much they had learned. This also supports the observation of the "variation" characteristic.

In line with Hargadon and Beckhy's (2006) findings, the Requirement Project Members' positive experiences "reinforced" the behaviour of supporting and contributing to the "reformulation of the problem". It gradually strengthened the "interactional frame" (Sawyer

2003) or the interaction pattern. This includes the way the Requirement Project Members interacted during the workshops, what tools they used, what roles the experts and the Requirement Coordinator enacted, how they communicated, and how they added to, and used, their respective domain skills. The process that shaped the interaction pattern can be described by the concept "collaborative emergence" (Sawyer 2003), since it was, to a great extent, jointly created by the different participants over time in a bottom-up process, and was not imposed from above by the project management. Once it was created, the frame started to restrain action alternatives and diversity. For example, the ultimate requirements had to fit in with the general data structure and graphical user interface. The interactional frame controlled what roles people could enact (no one suddenly tried to enact the role of facilitator for instance), and what initiatives individual members could take. This signified that the frame had a casual impact on the participating individuals, and contained the interrelated process known as "downward causation".

Gradually, the CH:s "restart" and "open up for diversity" and the new KI Apparatus with its sequential method, more planned meetings, standardization of documents, emergent interaction, time focus, and agents created what Okhuysen and Bechky (2009) refer to as 'integrating conditions for coordination'. At least the "predictability" and the "common understanding" conditions were present. Predictability in the sense of "having a sense for what subtasks make up larger tasks and in what sequence tasks will be performed" (Okhuysen and Bechky 2009 p.486) and common understanding in terms of sharing "knowledge of the work that is to be done, how it is to take place, and the goals and objectives of the work" (Okhuysen and Bechky 2009 p.486).

The outcome of the CH:s and the new KI Apparatus was the reformulation of the problem and project task. From a problem-solving perspective, project members used a too simple a problem-solving technique (Lyles and Mitroff 1980) during the first year. A problem-solving heuristic was used, similar to a "simple search" or "hill climbing" strategy (Dunbar 1997), where the project members (both the requirement members and the developers) "randomly" took one step at the time since they did not have the overall picture of the task or problem in mind or a more sophisticated search method to use. This resulted project members' failure to represent the problem in a solvable and beneficial way and delimit a reasonable problem space (Newell and Simon 1972; Dunbar 1997). In line with Dunbar's (1997) reasoning, the CH:s that were invented can be looked upon as rules of thumb or problem-solving strategies, as explained in the previous chapter (6), that constituted a direction and potentially successful way towards the end goal. But there was no guarantee that the CH:s would work. Heuristics are crude concepts that are not specified or described exactly – which is also the point with them; a heuristic should be quick and easy to use, and get its form during its application and in relation to an actual situation or problem.

The KI Apparatus that was used in this phase, the KIA 'Sequencing and Group Problem-solving' in the model, thus changed through the different CH:s, The CH:s were the drivers in the development process, resulting in "actionable knowledge" (Cross and Sproull 2004). Because of this, the project could progress and search for a solution to the reformulated project task.

3. The CH:s and the KIA in the Design and Programming Phase

The 'Design and Programming' stage of the process involved the emergence of new problems and attendant CH:s and changes in the KIA. In the following discussion, we will first see what

KI mechanisms were implemented then the most important KI agents in this phase are presented followed by a discussion of the knowledge integration activities and communication activities that took place in this stage. This section concludes with some reflections on the development process in this phase.

3.1 KI Mechanisms

New knowledge integration mechanisms, activities, and processes were implemented when the project was restarted and the process was opened up for more diversity. For the developers, the CH:s 1 and 2 involved a change in the KI Apparatus which led to the sequencing of activities within the frames of a cross-functional project, detailed project plans and schedules, milestones, tollgates, the formalization of handovers, standardized documentation, formal weekly follow-up meetings, and more rules and routines, directives and control mechanisms, such as the "project police". These tools can be seen as different KI mechanisms, similar to the means that Grant (1996a) discusses as "sequencing", which means that tasks are organized and carried out independently one at a time, in a specific order, and "rules and directives", which includes the use of policies and procedures, plans, and schedules. Formal routines (Grant 1996a) were introduced to predict and direct behaviour with respect to the delivery of functionality, and the error and change procedure. A bureaucracy was established to stabilize the development project and its process, which framed and steered development activities and hindered individual actions. Through these measures, the space for individual creativity was limited.

The overall programming task was decomposed into sub-tasks which were assigned to different developers. This can be compared to the KI mechanism "modularization" that Schmickl and Kieser (2008) argued constitutes one way by which the need for knowledge sharing and face-to-face communication can be reduced. Thus, the design and programming work depended less on the KI mechanism "group problem-solving" (Grant 1996a) in comparison with the period before the restart, and in comparison with the requirement work.

In line with Söderlund (2010), these knowledge integration mechanisms, which also were associated with the creation of a priority list and a "good enough-maxim", were used by the project management to orchestrate, pace, and synchronize knowledge processes that previously had been uncoordinated and thus hindered knowledge integration. Together with a heavy workload, this reduced the developers' individual creativity, the possibility to generate new solutions, or improve existing solutions. This restricted deviations from the plans. The developers did not care much about time pressure and time plans per se, but the ELWIS Programming Leader found that one way to reduce undesired creativity in these individuals was to overload them with work and offer them new tasks before they had finished ongoing activities (as will be explained more in 3.2 "KI agents").

Even though the mechanisms that were implemented by the CH "restart" brought some order and facilitated the structuring and organizing of the development process and its different activities, not all collaboration and integration problems were solved. In fact, new problems appeared. The initial unclear technical direction, problem 3 in the model, and the roles and relationship between designers and developers needed further clarification, for example. Designer and developer conflicts negatively influenced the development work within IT part of the project, and the communication with the requirement members. To solve the direction problem, the role conflicts, and the communication difficulties with the requirement members, and to further strengthen the KI mechanism "sequencing", a new CH (CH 3), called "create a navigational aid", was invented and implemented. It led to the formation of a separate design group consisting of three people and a new process step; the creation and documentation of the overall design. The design work was supposed to be completed before the programming work started. When the developers programmed the details of the system, they needed to consult the design group and the design documents to get a direction in order to align their contributions with the whole. This clarified and improved the relationships amongst the developers. In addition, it made it possible for the developers to show the requirement members how they intended to solve the problem. The CH, "create a navigational aid", can thus be interpreted as a complement and reinforcing factor to the KI mechanism "sequencing" (Grant 1996a). However, the designers' creation of the overall design documents entailed less room for developers' creativity and ideas, which the developers strongly disliked. The new directive to document detailed design solutions was often neglected. The CH "create a navigational aid" thus reinforced the diversity reducing effect by the formulation of the design at two levels, where the overall design framed what could be included in the detailed design.

In spite of the encompassing KI Apparatus, ambiguity remained in the development work, since different developers used different development practices and did not consider the KI mechanisms and management tools (sequencing, rules and directives, project plans) as being important or relevant to their work. In this situation, the KI mechanisms needed to be complemented with yet another CH – "bound creativity" (CH 4) and further changes in the KI Apparatus, consisting of less sequencing of overall- and detailed design activities, new KI agents, and changed forms of meetings and communication. These are discussed in more detail below.

3.2 KI Agents

The CH:s "restart" and "create a navigational aid" involved a new KI agent, the Design Leader (DL). The Design Leader role can be seen as a KI agent, since the job included the integration and matching of requirements and overall system solutions, as well as the integration of different sub systems. The Design Leader was also interested in the test process and in general management issues, which resulted in this person's involvement in all different parts of the project. This person can be characterized as having the same traits as Berggren et al. (2011) and Iansiti (1995) used when they described KI agents, namely, deep skills in one task-relevant domain and a general interest and broad skills in the rest of the stages of the development process and the product.

One problem that was encountered in the Design and Programming phase was that some of the leaders were distrusted by the developers. The developers argued that certain leaders were not capable of proving solutions; the need for stronger technical leadership still remained. Detrimental interpersonal conflicts emerged, called problem 4 in the model, which was solved by the CH 4, "Bound Creativity". The CH "bound creativity" brought about one new KI agent, the ELWIS Programming Leader. This CH also strengthened the importance of the Head Project Manager, who took the IT-subproject leader role in 2004. The new leaders, along with the strong directives to prioritize time and speed, and "good enough" solutions, were the most important elements in the KIA apparatus during this phase.

The ELWIS Programming Leader's KI integration capabilities were manifested in several ways. First, he succeeded in steering the group members in the same direction, and coordinated their previously disparate actions. While some diversity and individual differences and preferences were allowed (e.g. temper, manner, and preferences in working alone or together), “excessively” creative individuals were treated with a work overload and constant tasks in order to minimize the time that was available to explore new opportunities and develop new pieces on their own. Developers could not afford to “drift and jump” (Matusov 1996), completely unstructured, as they came up with new ideas or identified new possibilities, because striving for continuous improvement and perfection in a “disorganized” manner resulted in too slow a progress, and, in fact, inferior solutions at the overall level. Second, as was indicated in the previous chapters 4-6, the ELWIS leader collaborated and integrated his knowledge with an expert tester – a collaboration that radically improved the previously inferior quality of the system, according to several project members.

Third, the Programming Leader, the Design Leader, and the Head Project Manager possessed a KI ability that differed from, but can be added to, the characteristics that Berggren et al. (2011) and Iansiti (1995) have emphasised, namely, that they were able to act upon and integrate various experts’ tacit knowledge. The Design and Programming Leaders and the Head Project Manager used and listened to emotional reasoning and metaphorical speech in order to elicit tacit knowing and evoke discussions on important problems and issues that needed to be solved quickly. The leaders and managers put the information together with other information sources on a daily basis, which made them capable of quickly comprehending a problem and integrating relevant knowledge. They could, make decisions about what to choose and how to act and proceed without learning all the technical details. Emotional values, inner task-related intuition, metaphors and analogies were used as “communicative markers” that indicated how the project progressed.

Leaders with less impact on the developers’ interaction and production of work were thus replaced by stronger leaders who could stop discussions, make decisions, integrate contributions, and bear the consequences for different actions. In addition, the employment contracts of some strong individuals who made the integration of action difficult were ended. This action reduced the diversity in the group, as well as conflicts, discussions, and uncoordinated action.

3.3 KI Activities and Communication

At the end of 2003 and at the beginning of 2004, the Head Project Manager implemented measures, the CH “Bound Creativity” that resulted in changes in the KI Apparatus. This consisted of less sequencing of overall- and detailed design activities, and a change in the way programming leaders and developers communicated. This is explained below.

The change in the design work implied that the overall- and the detailed designs were performed simultaneously by designers and developers together. In this way, one may say, using Grant’s (1996a) terminology, that the sequencing of activities was lessened in favour of more group problem-solving. This was, nonetheless, done in a controlled manner and the documentation of the design at the two levels was still required. The new collaboration form reduced the time and technical gaps that had emerged. The developers’ ideas and knowledge were also better taken care of and integrated into the design solutions. The closer collaboration between designers and developers increased the number of possibilities where divergent opinions and unique knowledge could be expressed. As a result, the whole system, from design specification to the code, was radically improved.

The meetings from this time onwards mostly concerned actual problems and different action alternatives, whereas the plans and activity schedules were only hastily examined. The project kept the weekly status meeting time and place, but the "new" communication pattern was more problem-oriented. In addition, problem meetings were set up in, between the status meetings, as required by the emergence of new and unexpected problems. In Teasley's (1997) terminology, the discussions contained more "transactive reasoning" in comparison to the clean-cut status report meetings that took place in 2003, which could be characterised as being mostly 'non-transactive'. The space for diversity was enlarged through the change from status report meetings to problem solving sessions where different perspectives and options were analysed and tacit knowledge more easily tapped through emotional markers, metaphors and analogies.

In summary, with respect to the work with the design and the programming of the system, the most important knowledge integration measures that were implemented in this phase of the project were the project directives and rules, the employment of new IT-leaders (the ELWIS Programming Leader and the IT-manager/the Head Project Manager), and the CH:s "Create a navigational aid" and "Bound Creativity" (the priority list, the time orientation, and the "good-enough" motto). Once more inspired by Okhuysen and Bechky (2009), one may say that the directives, rules, and the CH:s together created integrating conditions in terms of accountability, predictability, and the common understanding that is necessary for knowledge integration to occur. Accountability had to do with making responsibilities among members clear, and predictability was associated with "having a sense for what subtasks make up larger tasks and in what sequence tasks will be performed" (Okhuysen & Bechky 2003 p. 486). Common understanding involved knowledge about the work and its objectives. As the integrating conditions evolved, the project leaders' task to pace, orchestrate, and synchronize actions was facilitated, as expressed in Söderlund's (2010) terminology. When the knowledge integration process started to be successful, the project managers could engage more in up-coming challenges and forward-looking problem-solving, instead of being occupied with analyzing what went wrong in different situations and solving acute problems that already had emerged.

3.4 Reflections

Before the developers and the requirement workers had found a successful way of working and a problem-solving strategy that could take them to the "goal state", they applied a simple search heuristic similar to "hill climbing" (Dunbar 1997) in the sense that they formulated requirements and developed system solutions almost one by one. These requirements and solutions had to be amended over and over again. The KI tools "sequencing", which included the "freezing" of documents and formal "handovers", and the CH "navigational aid", which entailed that different solutions were documented and tried out on paper to see how the pieces interrelated before they were programmed seemed to be a more beneficial long-term strategy in this project. The documentation enabled the participants to vary the solution on paper before the parts were actually programmed, and, in this way, different project members' way of thinking was more easily accessible to others. They could, together, better see how they were to build on each other's contributions. Once more, we can see how the KI Apparatus and the CH:s complemented each other, and made a working knowledge integration process achievable.

In the Design and Programming stage, the importance of "actionable knowledge" (Cross and Sproull 2004) could be recognized in terms of the "good enough-motto", the priority list, the time management, and the focus on solutions that served a practical purpose. The project management team instructed the developers to only correct and change things that constituted errors or bugs; no pure technical advancements were allowed. However, discussions among developers and testers and requirement members often turned to who was in the position to judge what was wrong, whether things were done properly, and where the error actually was rooted. Developers did not give up on their ideas easily, sometimes not even when they were disconfirmed. Instead, developers criticised the test cases and how things were tested, who carried out the tests, what was prioritised in the tests, and how the error reports were written and formulated. This is similar to Dunbar and Klahr's (1988) discussion on problem reformulation, confirmation bias and difficulties to abandon disconfirmed hypotheses and solutions. Nevertheless, epistemic contestation (Faraj and Xiao 2006) and disagreement (Matusov 1996) seemed to stimulate change and improve solutions. For instance, the developers did not appreciate the new KI apparatus with its sequential and formal way of working that involved more management control, documentation, time pressure, and "good enough" solutions at the expense of technical perfection. This disagreement may have been, in line with Matusuv's (1996) idea on disagreement as a prerequisite for development and learning, something that spurred further organizational change, problem solving and development since the developers' negative response to the sequential method induced a change in the way designers and developers collaborated. This "bottom-up" change improved the quality of system solutions drastically, as described above, and people became satisfied with their work situation.

4. The CH:s and the KIA in the Test Phase

In the test and evaluation stage, new problems and CH:s were encountered, which once more had an impact on the KI Apparatus in use. This is described below. As a final reflection on this process, it is discussed how a specific "selection or concluding mechanism" that existed in several CH:s (but not in the KI Apparatus) throughout the development process seemed particularly important in this project.

4.1 KI Mechanisms

The old component test conflict that existed between the line units "test" and "developers" with respect to how much should be tested by whom, where, when, and how was inherited by the project and became even more heated during the project process. Since the existing sequential test process did not resolve this problem (number 5 in the model), the project members invented and implemented a "new" temporary strategy and CH (CH 5 in the figure); "Upfront Substantiation". As has been explained before, this CH involved close collaboration or "group problem-solving" in Grant's (1996a) words; here between the ELWIS programming leader and a skilled tester. The solution solved the conflict and the problem for the project, and is interpreted as a successful change in the KI Apparatus from "sequencing" to "group problem-solving".

In general, the KI mechanisms that were used by the test group were sequencing of test activities, PPM's universal test methods and rules, and standardized test procedures regarding, for example, test cases, test tools, test data, test environments. These activities can be likened to the "sequencing" and "rules and directives" and "standardization" mechanisms that Grant (1996a) and Kellogg et al. (2006) have discussed. The standardized operations, documentation, and communication forms also involved some modularization (Schmickl & Kieser 2008) of the task in the sense that different subsystems' component tests were carried out independently by different testers. The work was divided in accordance with the different systems; two testers worked with the ELWIS system, two testers with the K2 system, and two other testers were assigned to the testing of the communication channels. When each subsystem had been tested, a pair of testers integrated the systems and tested information flows and how different systems communicated and processed information together. This made it possible for the testers to work "alone" or in minor teams and pairs, which in turn lessened the need for whole group interaction and interpersonal communication.

However, there was not enough time at the end of the project process to sequence test activities and work independently to the extent that was planned for. As mentioned earlier, this project was ruled to a great extent by "time" as a KI mechanism (Söderlund 2010, Okhuysen & Eisenhardt 2002, Tyre et al. 2002, Gersick 1989), and when standardized line operations met project intentions problems arose. Testers argued that there existed a universal PPM test process that the project wanted to depart from. However, at some point in the project life, the managers and testers realized that there was too much left to test in the light of the time that was available. In addition, there were too many faults left and probably even more errors that were still unidentified but must be corrected before the system could be implemented in the organization. The testing and confirmation of the system went ahead too slowly (problem 6 in the model). There was a great sense of urgency at this point in time, as explained in Chapter 4. Tests had to be run in parallel and people had to interact more, both within the test group and with other project members. This situation reminds one of Lindkvist's et al. (1998) study of a product development project. They found that deadlines and time pressure were critical, not only to hasten the development pace and deliver the product on time, but also to encourage cross-functional communication, to carry out different project activities in parallel, and to coordinate action and make "good-enough" decisions from a "holistic" point of view. The existing KI Apparatus (sequencing, standardized test operations, and rules and directives on how to carry out tests at PPM) showed itself once again to be insufficient.

This state of affairs resulted in the implementation of CH 6 "cross-check", as explained earlier, which involved cross-functional pair-testing, and increased interpersonal interaction and communication. At the end of the project, several project members argued that this was a good way to speed up the process and they said that they could effectively utilize and combine each other's competence when they worked together in front of the computer screen. In addition, morning meetings were scheduled to make the work and actual progress "visible" for all of the testers. It resulted in improved knowledge integration in this stage of the process, as suggested by advocators of "free interaction and dense communication (e.g. Wheelwright and Clark 1992; Brown and Eisenhardt 1992; Pinto and Pinto 1990). The need for more "group problem-solving" in the test phase can be contrasted with the previous stage where less informal interpersonal communication and interaction was needed to increase knowledge integration performance.

4.2 KI Agents

As mentioned earlier in this chapter, the important KI agents in the test process included one of the Programming Leaders and the externally recruited test person who, in collaboration with each other, integrated their diverse but complementary knowledge (programming and system skills and deep technical test knowledge, respectively) to test the system in its early phases. They managed to improve the quality of the system considerably.

In addition to these two persons, the Operations Subproject Manager (OSM in the model) played a crucial role as a KI agent when she initiated an improvised change in the organization of acceptance tests, consisting of new collaboration forms and a new test strategy. This was described above and earlier as the CH "cross-check the system". End users were paired with testers, and, together, they combined their knowledge and divergent perspectives and performed more discovery-like tests. The important agents here were thus not only the "ordinary" testers, but also the users who assisted and collaborated in the test process.

There were also some other technical test experts who were important to the knowledge integration process. These experts participated in detecting and solving interface problems between different subsystems and between the systems and the technical infrastructure. In addition, the project members who participated in the change advisory board discussed and solved problems that had an impact on different knowledge domains. These members had some general or broad understanding of several different parts of the development process, but also deep skills in one area, similar to the characteristics that Berggren et al. (2011) emphasize, and Iansiti's (1995) T-skills pattern.

4.3 KI Activities and Communication

Even though there were an number of important individuals in the test and evaluation process, the most crucial actions were carried out in small groups; the upfront substantiation, cross-checking, the advisory board meetings with continuous collective pragmatic reasoning process, were all collaborative processes. These processes took form through the creation and implementation of the CH:s "restart", "substantiate up front" and "cross-check the system", which in turn came about when the project experienced difficulties which the existing KI Apparatus could not resolve. The change in the KI Apparatus that the CH:s brought about in the test and evaluation phase generally involved a switch from sequencing, standardization, and rules and directives (Grant 1996a) to more group problem-solving (Grant 1996a).

Using Enberg's et al. (2006) notion on 'integrating knowledge alone versus together', one may say that the CH:s "substantiate up front" and "cross-check" implied more collaboration and joint integration of knowledge, and less integration of knowledge alone. The fundamental reason behind more collaboration and communication was to improve and finish the system more quickly to shorten the total test time, and to reduce the number of iterations. The "substantiate up front" which involved the Programming Leader of ELWIS and the test expert, and the "cross-checking" which entailed end users and testers' collaboration, were more fruitful than when the same individuals worked alone or in their functional groups. The existing KI Apparatus did not stimulate or enable a re-organization of the work, nor did it and resolve the problem of failing to detect faults in a timely manner.

By using the CH "cross-check", a new form of testing was introduced, where testers worked together to test the system in an exploratory-like manner. Here the strategy was to put less effort into conducting pre-scripted tests or holding on to what had been "pre-selected" (Lindkvist et al. 2011) as relevant trials, or what was right or wrong in this context. Instead, the testers were tasked to "drift and jump", using Matusov's words (1996), and to broaden their perspectives and keep their eyes open for unexpected things. Even though pre-determined written test cases were still used, there was not as much focus on them in this strategy as before. Tests were carried out more in an improvised manner, and different tests were carried out in parallel. Consequently, progression occurred in a "mosaic fashion" (Matusov 1996); different pieces were tested and put together non-linearly by different people working from different angles. Furthermore, parallel tests and discovery-like testing, which involved "expect the unexpected", might be understood as a shift in problem-solving or heuristic strategy towards a more "simple search" heuristic (Dunbar 1997), where the project members took one step at a time and analysed what was found before they decided on where to go next. This is similar to Magnusson and Lakemond's (2011 p.144) finding on knowledge integration processes in new product development; "it seems that later in the project there is still an increased need for knowledge integration characterized by iteration and experimentation".

With respect to the test and evaluation activities, diversity was allowed more space after the implementation of the CH:s. In both of the CH:s that were connected to the evaluation and test activities ('substantiate up front' and 'cross-check the system'), people with different kinds of knowledge were organized to collaborate in cross-functional settings. The KI Apparatus that was used initially indirectly involved diversity reducing forces, which were counteracted by the CH:s. For example, a test team constituted a separate functional group instead of a integrated cross-functionally organized team. This was changed with the CH:s, since the "upfront substantiation" included close collaboration between tester and developer, and the "cross-check" entailed close collaboration between testers and end users. The CH:s increased group problem-solving, which constituted a change towards more diversity in the KI Apparatus. Moreover, the KI Apparatus that was used initially mostly involved pre-scripted testing that was based on requirements and design documents, and constituted a kind of standardized test procedure that testers' were expected to follow. In turn, this limited spontaneous creative thinking. This situation changed, as explained above, when the testers and the end users started to "cross-check" the system using a more discovery-like strategy. In addition, the sequential model in itself implied that the testers, who were the last instance in the process, had less influence on the product as a whole, a situation which one of the CH:s ('upfront substantiation') could only influence to a limited extent.

One standing issue during the Test and Evaluation phase was whether testers should or should not keep a distance from the systems that they examined and the developers who created them. Here, the testers and the developers had different beliefs rooted in their knowledge differences on how to best test and evaluate the system solution. This disagreement can be compared to Faraj and Xiao's (2006) discussion on the practice of "epistemic contestation", which concerned patient treatment and safety conflicts between specialists from different medical domains in a trauma centre. Testers argued throughout the process that there was a point in maintaining a certain distance since they then were better able to test the systems more "impartially" or "objectively". The developers, however, argued that the testers had to come close to the system and learn about the system from the "inside", so as to be able to carry out valuable tests. Writing test cases, which constituted one important part of testers' work, was an effort to create a fallible version of the system and framing the "test problem". The creation of skilful test cases demands that the testers knew about the requirement and design specifications. They had to check that all the requirements had been met that the solutions matched the requirements exactly. It also placed demands on the testers' creative thinking in cases where they were challenged how to systematically identify the systems' weak parts and identify bugs in the system. The testers would then force the system those things go off. In the testers' "thought world" (Dougherty 1992), the more errors and weaknesses a tester found, the better tester he or she was, which the developers sometimes argued was an exaggerated characterization. However, taken together, this meant that test cases included ways to both confirm the system and crash the system. The second strategy demanded more of the testers in terms of creativity, system knowledge, and business processes behind the system, and, presumably, deep "inner" system knowledge. At the same time, to be able to criticize the system and pinpoint its weak parts, necessitated in the testers a certain emotional distance and differentiated thinking style with respect to the system.

Irrespective of what was right or wrong, disagreement itself worked as a quality enhancing factor, in a manner similar to what happens between a defender and lawyer in court, as Matusov (1996) has described. The developers' attempts to make testers more focused on the system's inner construction and the testers' simultaneous ambition to stay outside the development of the system whilst still possessing tacit knowledge of the system, might have constituted a factor in the development process which took the product to higher quality levels than would have been the case if the developers and the testers had shared beliefs and attitudes. This is because the disagreement between the two groups resulted in new forms of collaboration.

4.4 Reflections

At an overall level, the whole test activity can be seen a "selective retention process" (Lindkvist et al. 2011), since it aimed at assessing the system and determining its function and value. In this project, selective retention was carried out, not only by the different tests per se, but also by different supporting tools such as correction and error handling procedures (error reporting system, classification of errors in terms of importance, consequence and priority assessment).

The error correction procedures constituted a "trading zone" (Kellogg et al. 2006) since they entailed different exchange rules. The testers tested the system, but could not always determine the technical implications of their findings. However, they were still able to communicate their findings from the tests to the developers in a understandable way. Consequently, the developers could carry on the correction work and then resend the improved system solution back to be retested. Even though the error reporting system contained different defects, according to both testers and developers, it still supported the process in a more coherent way than compared to the time before the restart of the project.

The change advisory board meetings were also important to the "selective retention process". The board consisted of the project management and different experts. They met regularly to discuss the nature, consequences, correction time, and importance of the bugs, and made decisions on whether to correct the bugs or not. On these occasions, the project members engaged in collective "critical inquiry" (Lindkvist et al. 2011) which was aimed at keeping control of the progress so that different individuals did not offer their own interpretations of the errors or decide independently what to do with the errors. The practical value of the error for the future users was an important evaluation and decision criterion in these discussions. This constituted a selective or integrative communication aspect that was applied in order to identify what was important and had to be corrected and changed in the system before its implementation. The final reflection in this section relates to this.

The last reflection concerns the importance of the user involvement and the pragmatic perspective that several project members and the project management maintained throughout the process. This perspective was visible in the requirement phase, in the design and programming, and during the testing and evaluation of the system. The CH:s "restart" and "open up for diversity" in the problem formulation phase involved user participation. The CH "create a navigational aid" in the solution search phase implied that future users could see and understand in advance what the solution would look like, and the "bound creativity" in the same phase entailed a priority list that emphasized the *user* requirements and “good enough” principle. Finally, the CH “cross-check” in the evaluation phase included end user involvement. The pragmatic or practical reasoning that was evident here was a way to bring the different pieces of the system together and finalize the system. The KI Apparatus did not entail pragmatic-oriented tools, but the CH:s did include such mechanisms. What counted was what worked; things should fulfil a purpose and correspond to a need since it was to be the end users’ acceptance or denial that eventually determined whether the project was successful or not.

5. Summing up the relation between the CH:s and the KIA

A variety of KI tools could be identified in the different phases of the project. The most important ones in the requirement phase were sequencing and group problem-solving. In the design and programming phase there was more emphasis put on finding strong knowledge integration agents and establishing a number of crucial directives on how to work and what to prioritize. In the test phase, more group problem-solving could be identified again. However, it is important to note that the KI Apparatus was not completely changed in each phase; some elements persisted throughout the entire process.

The KI Apparatus offered different organizational practices and solutions, but could not easily respond to unexpected problems. Once implemented, the KI Apparatus seemed to constitute and work as a static organizational “structure” with ingredients that did not easily change, without inducement. In addition, the KI Apparatus did not seem to sensitize diversity-related problems and could therefore not be used to adjust the diversity space appropriately over time, in different situations. This is where the CH:s come in. They emerged because of at least three reasons. First, the CH:s complemented the KI Apparatus and solved problems that the existing KI Apparatus could not manage alone. Second, the CH:s initiated changes in the KI Apparatus to adjust to the actual situation. Third, the CH:s assisted the KI Apparatus in regulating and fine-tuning the space available for expressing and using project members’ task-related diversity and skills throughout the process. The CH:s could, therefore, be seen as dynamic responses to current situations and problems. This is however just a tentative hypothesis concerning how the CH:s may allow for new sets of KI mechanisms and processes. The CH:s contributed to or initiated changes, but that is not the same as claiming that they can explain the every change that occurred during the lifetime of the project’s development.

CHAPTER VIII

VIII. DYNAMIC KNOWLEDGE INTEGRATION
Conclusions and Contributions

In this concluding chapter, I begin by reiterating the purpose of the study and the research questions that specify the purpose. Each of the research questions are responded to. In section 2 below, I present a model of Dynamic Knowledge Integration that summarizes the findings and contributions of the present study. Finally, in section 3, a number of ideas for future research are suggested.

1. Purpose and Research Questions

As stated in the introduction of this thesis, *"the purpose of the study is to understand how project groups manage the process of knowledge integration in a complex development context which is characterized by expertise diversity and a limited amount of time with which to complete the task on hand. It seeks to provide understanding of how project members use various knowledge integration mechanisms over time to integrate knowledge and strike a balance between making the most of project members' heterogeneity of expertise and at the same time meeting the demand for time-efficient coordination and swift progress. This will involve the identification of different knowledge integration mechanisms for modulating the space that is available for expressing individual differences. It will also include a description of how specific knowledge integration mechanisms may vary over time. In addition, the study presents an examination of how project members themselves invent and use a variety of helpful measures to facilitate progression at different stages of the development process. The general ambition of the study is to contribute to current research on knowledge integration by adding a process understanding of the utilization and creation of knowledge integration mechanisms in complex development settings."*

This purpose was operationalized by two more specific research questions which are discussed and answered below.

The first research question was: "*What knowledge integration mechanisms and other collective means to enable knowledge integration and project progress can be identified throughout the project process?"*

Identified and explained in the previous chapter, we see that the traditional KI Apparatus, including mechanisms, agents, activities, and communication consisted of *sequencing* throughout all of the three problem-solving phases, although it was reduced in the third phase in favour of a more parallel-like test method. *Group problem-solving* was applied in especially the first and the last part phase, and in a structured, more rule-based and disciplined way in the design and programming phase. In the design and programming phase and in the beginning of the test period, some group problem-solving was replaced and limited by *modularization* principles. Furthermore, *time pressure* as a knowledge integration mechanism and project management tool was an apparent factor throughout the entire process. Moreover, *standardization*, including the documentation of requirements, overall design, detailed design, handovers, code deliveries, and test and change procedures, was emphasised from the restart of the project. This was done in order to align and integrate actions in all of the stages of the development process. Standardization was connected with documentation and traditional *project management tools* such as plans, milestones, meetings, and activity schedules, as well as *rules and directives*.

As I revealed in the discussion in the previous chapter, there were some important KI agents with deep skills in one area and broad skills in several other related domains. The Requirement Coordinator, the Design Leader, and the Head Project Manager belonged to this category. Additionally, end users, different experts, Programming Leaders, and the Operations Sub-project Manager played crucial roles in the knowledge integration process.
A salient pattern with regard to knowledge integration activities and communication was that spontaneous and free communication needed to be more structured in order to facilitate and enable the integration of knowledge. In the problem formulation stage and in the test and evaluation stage face-to-face communication and cross-functional collaboration was increased and in the design and programming phase face-to-face communication and cross-functional collaboration decreased. However, in all cases face-to-face communication and cross-functional collaboration became more organized. Moreover, even though the general development method consisted of sequencing of activities to a large extent, there was room for more "emerging collage-like", "mosaic crafting", non-linear work, and simultaneous or parallel work within and between each phase of the development process.

One element of the research question, *other collective means to enable knowledge integration,* consisted of the collective heuristics that were invented by the project management and members in order to solve different unexpected problems. These were problems that could not be solved within the frames of the existing KI Apparatus. The collective heuristics complemented the KI Apparatus, initiated changes in the KI Apparatus, and adjusted the space for expressing and utilizing diversity appropriately during the process. Each collective heuristic was connected to a specific problem, as previously described in detail, but contained general aspects which may well be useful in other similar development contexts.

The collective heuristics were also identified and associated with specific stages of the development process, but some of them overlapped to some extent and may appear in other phases as well. In the first stage, problem formulation, the collective heuristics were "restart" and "open up for diversity". These heuristics were a response to a poor understanding of the task at hand, inappropriate or too narrow a problem formulation, and insufficient integration of different experts' knowledge. In the solution search phase, the collective heuristics were "create a navigational aid" and "bound creativity", which were related to problems of misalignment of actions and "wild" creativity. In the third stage, test and evaluation, the collective heuristics consisted of "substantiate up front" and "cross-check", which responded to problems of inferior system quality, which, in turn, was rooted in cross-functional collaboration problems and a lack of time to continue testing "by the book" in accordance with the KI tools in use at the time. The collective heuristics were connected with collaboration difficulties and solutions among diverse people who, seen from a problem-solving perspective, jointly tried to identify a problem, search for a solution, and test and evaluate the result.

The second research question was: "*How are the identified knowledge integration mechanisms and the other knowledge integration enablers used to extend or limit the space used for expressing knowledge differences in different project phases – in the face of the approaching deadline?*

As explained in the previous chapter, the collective heuristics modulated the space that was available for the appropriate expression of diversity over time. In the first stage, the space for expressing differences was enlarged, in the second it was mostly limited and in the third phase it increased again.

In the first phase, the collective heuristics “restart” and “open up for more diversity” were used to extend the space for diversity by enabling simultaneous integration of different expertise in properly organized workshops and seminars. This was done in order to reformulate the problem and create a comprehensive requirement portfolio. In this phase, diversity was kept under control by the integrative conversation pattern that unfolded during the sessions. The development process and the project members lacked knowledge integration tools during the first year of the project, and diversity could thus not be effectively managed. This was changed by the implementation of the CH:s and the KI Apparatus that was introduced by the restart of the project. It involved different means, such as group problem-solving, knowledge integrators, cross-functional collaboration, help seeking and help giving behaviour. These different means facilitated the promotion of diversity. There were, however, diversity disciplining forces, such as time restrictions, standardization of documents and changes, the Requirement Coordinator’s role, and cross-boundary communication that kept actions coordinated.

In the second stage of the project, the collective heuristics “create a navigational aid” and “bound creativity” restrained the possibilities for the expression of individual creativity and diversity. The developers were asked to create a design solution that was to direct actions and keep different developers’ ideas and solutions within a certain frame. In addition, “bound creativity” involved various tools, such as a priority list, the ‘good enough’ motto, fewer free- and open discussions, new stronger leaders which were used to bring an end to the generation of disparate solutions. Individuals’ widespread preferences and opinions on what was needed were also suppressed. The existing KI Apparatus, which to a large extent was based on group problem-solving or self-organized developing teams, resulted in detrimental conflicts, indecisiveness, and individual creative freedom at the expense of collective performance. It permitted too much diversity, which was later reduced by the CH:s. The CH:s also triggered changes in the KI Apparatus, which came to include more plans, activity schedules, rules and directives, a sequencing of activities, powerful leaders, and modularization. These CH:s were implemented to reduce interpersonal communication, and to promote an even stronger time focus. These were all ingredients that helped restrict the space for the expression of dissent and diversity. However, since the project members’ deepest skills and knowledge were needed, there were also factors that stimulated the appearance of unique knowledge. The factors included a change from status report meetings to more problem-oriented sessions, less sequencing and handovers of overall- and detailed design documents in favour of more organized group collaboration, and the communication of tacit knowledge in the form of task-related emotions, and metaphorical and analogical expressions.

In the test and evaluation phase, the CH:s were mostly used to extend the space for diversity. When the first functionality package was tested, the project members realized that the inferior system quality was worse than anticipated. This problem was solved by closer cross-functional collaboration, thereby enlarging the space for diversity. The aim was to detect errors and substantiate the system early. The existing KI Apparatus, with its sequential model and formal handovers, could not accomplish this change and solution since it kept experts, that is, the developers and testers, apart from each other and so reduced the space where differences could blend and result in high-quality solutions. Moreover, when time was running out and the project’s deadline was approaching, a new need to change the KI Apparatus emerged. The sequential, standardized, and rule-based test method minimized the room for creativity and diversity, and was replaced by the collective heuristic “cross-check” which extended the space for diversity by entailing a cross-functional (testers and end users) parallel search for unexpected errors and hidden bugs in a discovery-like fashion. However, it was important to progress quickly and predominantly focus on the correction of errors that were classified as significant and of highest priority. Here a pragmatic criterion was used, which involved judgement on what was useful and purposeful from, above all, a user perspective. This pragmatic reasoning process is understood as an integrative factor that narrowed the space for conducting individual actions and developing diverse improvement ideas.

2. Modelling Dynamic Knowledge Integration

My claim of a “research gap”, as described in chapter 2, referred to our need to study and enhance our understanding of knowledge integration as a process and to complement what we know about knowledge integration by looking into the dynamics of how KI changes over time in a specific development process. Furthermore, it was argued that more attention should be paid to the knowledge integrators, the doers, that is, those who conduct knowledge integration in their day-to-day business life. In addition, expertise diversity was acknowledged as a prerequisite for complex development, but, at the same time, it is a factor that has not been fully investigated in relation to knowledge integration in the sense that it needs to be managed appropriately during the development process. Expertise knowledge must be used when and where it is needed. In order to extend our general understanding of knowledge integration from these perspectives, an explorative-like case study was carried out and presented in this dissertation – a type of study that other researchers (e.g. Tell 2011) have called for.

The main outcomes of the thesis are the identification of the significant role that “collective heuristics” play in the knowledge integration process, and their relation to other knowledge integration mechanisms and activities. The collective heuristics complemented the KI Apparatus and initiated changes in the KI Apparatus and can thus be interpreted as “drivers” of the KI process. The interplay between the KI Apparatus and the CH:s offers us insight into the underlying processes of knowledge integration. It extends our understanding of knowledge integration as a process by explaining how such a KI progression unfolds over time, from start to end, within the frames of a specific development project.

Below, I reveal my theoretical positioning by presenting a figure that illustrates Dynamic Knowledge Integration. This is a tentative model that is intended to contribute to the field of knowledge integration. I will first shortly explain the process and the elements of the model before I go into the details of each part and engage in a more theoretical discussion. What I show and explain is the role that collective heuristics play in knowledge integration processes, and that the evolving CH:s and the KI Apparatus together form, what I call, Dynamic Knowledge Integration.

The large block arrow in Figure 12 symbolizes the development process. This process consists, in turn, of two interacting processes; the knowledge integration (KI) process and the collective heuristics (CH) process. The KI process consists of different sets of knowledge integration mechanisms, activities, and sub-processes, which have been discussed in the theoretical point of departure (Chapter 2) on *how* knowledge integration is enabled and conducted. The different configurations of mechanisms, activities, and processes are represented in the figure by the boxes called 'KIA' as in the previous chapter, Chapter 7. 'KIA' stands for Knowledge Integration Apparatus. The CH process comprises of the different collective heuristics (CH) that were invented to solve problems, adjust the space to express different perspectives and knowledge differences, initiate changes in the KI Apparatus, and to improve collaboration during the course of the project.

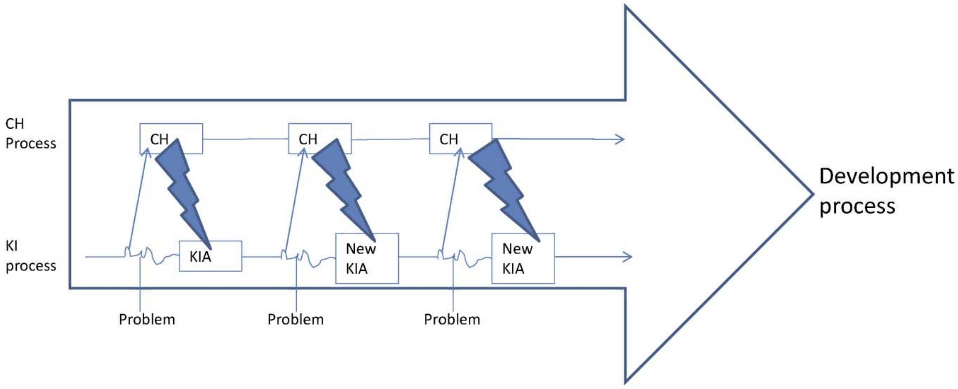


Figure 12: Dynamic Knowledge Integration

The model shows the case project's process and the interplay between the KI and the CH:s as it unfolded in the project over time. The development is ongoing when problems emerge, depicted by the first wavy part of the KI-line. The CH 1 and CH 2 were formulated, and a specific KIA, called KIA Sequencing and GPS, was implemented as a result of the CH 1 and CH 2. This is illustrated in the figure by the first blue lightning bolt. Order is restored, which is represented by the straight part of the KI-line. However, new problems emerge, which is represented by the next wavy line in the KI process. The existing KIA fails to solve the problems. A change is required. New CH:s were created, CH 3 and CH 4, to solve the problems and to implement a change in the KIA. This is represented by the next blue lightning bolt. Modifications in the knowledge integration apparatus result in the new KIA Agents, Rules and Directives. Order is restored again, which the straight part of the KI-line after the KIA box in the middle of the figure shows. Once again, the project members and management encounter critical problems, illustrated by the last wavy section of the KI process line. The current KIA cannot deal with the problems that emerge and a new change in the KI Apparatus is necessary. New CH:s, CH 5 and CH 6, are implemented, resulting in a new configuration of knowledge integration mechanisms, activities, and sub-processes. This is indicated by the last blue lightning bolt and the KIA 'Mosaic GPS'. Order is reestablished, which is denoted by the remaining straight part of the KI process line. No more problems emerged in this project, no additional CH:s were invented, and no further changes in the KIA were made. The interplay between the CH process and the KI process over time is here called Dynamic Knowledge Integration. It is an emergent pattern that I observed during the course of the project, but one should remember that it does not perfectly mirror reality. It is a simplified version that cannot completely explain the relationship between KIA and CH:s.

As an aid to conceptualize the pattern mentioned above, I propose that dynamic knowledge integration is a process that occurs at two levels. There is one underlying knowledge integration process that consists of different knowledge integration mechanisms and activities.

The other process that comprises different CH:s becomes activated when the knowledge integration process has been interrupted by problems that cannot be solved by existing mechanisms and activities. The latter process can be described as a dynamic capability (Teece et al. 1997), or mechanism of change, since it reconfigures coordination and integration activity and accomplishes a shift within the KI process that helps prevent stagnation. This can be compared to Teece's et al. (1997 p.515) definition of dynamic capability (even though they refer to firm-level processes whereas this discussion refers to the project level); "[w]e define dynamic capabilities as the firm's ability to integrate, build, and reconfigure internal and external competences to address rapidly changing environments."

However, as we can see, the process was not characterized by constant adaption or reconfiguration. In Gersick's (1988) terminology, the progression can be described as a pattern of "punctuated equilibrium". A pattern of punctuated equilibrium is featured by periods of continuity that are suddenly interrupted. The way of working, or in Gersick's (1988) theory, the "task approach", goes through a transition, which results in dramatic progression and a new period of equilibrium. In the present case, the transitions were activated by unanticipated problems and resulted in a changed form of coordinated activity and knowledge integration. New strategies, CH:s, were created to solve particular problems and the current KIA was abandoned or at least modified. The project members and managers maintained the new way of working for as long as possible; they did not wish to work in a constantly changing environment. The goal was to establish a working knowledge integration process and reestablish stability within the project. This can also be compared to the "project-led learning" in Brady and Davies' (2004 p.1615) "Project Capability-Building Model". In their study project members faced a market challenge that triggered a change in current project processes and way of working: "existing in-house project processes were unable to cope with scale and complexity of the new types of projects demanded by their customers. Where possible they relied on existing routines and capabilities, but had to develop new areas of knowledge and expertise to meet the changing requirements of their customers." (Brady and Davies 2004 p.1615). As in the current case, […] learning tended to be on an ad hoc basis […] and "[p]roject capabilities continued to be built from the 'bottom-up' by the project business organization". (Brady and Davies 2004 p.1616)

In Okhuysen and Bechky's (2009) terminology, we see that the knowledge integration process was steadily influenced by "everyday dynamics" and was repeatedly intruded upon by severe problems. This turned the knowledge integration effort into an "ongoing accomplishment" (Okhuysen and Bechky's 2009) which could not be achieved only by structural coordination mechanisms which are imposed as faceless organizational design and work principles (such as project rules, directives, development methods and plans) by the top management. There was, instead, a bottom-up dynamic where the project members and participating project managers and their actions played a prominent role. Some project participants enacted a role as knowledge integration agents (Andersson and Berggren 2011) and were of particular importance with respect to the occurrence of knowledge integration. When the more rigid KI Apparatus collapsed or was found to be insufficient to the task on hand, the project members managed to restore coordination and knowledge integration through the invention of different collective heuristics. This is an indication of the important role that the project participants and their situational and problem-oriented actions played. The CH:s reveal that KI is evolving and frequently more about "dealing with the situation" than about formal organizational arrangements", as Faraj and Xiao (2006 p. 1157) argue. The role that CH:s played and how to theoretically understand the notion "collective heuristics" is discussed below.

As Okhuysen and Bechky (2009) and Faraj and Xiao (2006) discuss, knowledge integration in fast changing environments cannot easily be pre-specified. It is an evolving process. CH:s are problem-solving strategies that respond to unexpected problems in the knowledge integration process and cannot, therefore, be defined in advance. Similarly, Eisenhardt et al. (2010 p. 1266) suggest that "[h]euristics emerge as individuals adjust to problem-solving situations in which there is limited time and information". Kahneman (2011 p. 98) states that "[t]he technical definition of *heuristic* is a simple procedure that helps find adequate, though often imperfect, answers to difficult questions. The word comes from the same root as *eureka*."

The concept 'heuristic' has been associated with different individual mental processes such as assessing the probability of different events, intuitive judgment, decision-making and problem-solving (Tversky and Kahneman 1974; Kahneman 2003; Gigerenzer 2008). Numerous heuristics have been identified and discussed in the literature on, for example, psychology, computer science, mathematics, and behavioral economics. Heuristics are often discussed as a means to solve complex problems, of which a complete analysis with all aspects, variables, details, scenarios, and possible solutions cannot be conducted by the human mind. Complete information and enough time might not even be available. Due to an individuals' limited capacity, or "bounded rationality" in Simon's (March and Simon 1993) terminology, decisions, choices, and problems must be simplified and narrowed so that they can be dealt with. Kahneman (2003) builds on this idea and discusses how biases and illusions are unconsciously used to make judgment easier. Dunbar (1997) and Gigerenzer (2008) discuss more positively how heuristics can enhance problem-solving and decision-making and even outperform detailed analysis. Polya (1945) suggests that different heuristics are conscious strategies which are used to try out when one experiences difficulties in solving complex math problems. Inspired by this observation, the present study identifies and positions collective heuristics as deliberate strategies, but potentially imperfect. They are strategies that are invented and applied to solve difficult problems that cannot wait for complete analysis but need quick action. The *collective* part of the term, *collective heuristics*, refers to the fact that the project's problems and solutions belonged to, and affected, the whole project team. The term thus emphasizes the collective process and the social dimension of problem-solving activities that take place within organizations.

Returning to organizations and the field of knowledge integration, I will further discuss the term, *collective heuristics*, by comparing it to some other organizational concepts. In line with Eisenhardt et al. (2010), one may say that as in contrast to routines, heuristics are flexible and temporary, and adequate for ill-defined problems. "Because heuristics are easy for organizational members to remember and are quick to use, they provide efficient guidance for some actions, but just as important, they also leave room for flexible adjustment in real time of other actions. Heuristics are thus distinct from routines that provide detailed and automatic guidance for well-specified problems and so favor efficiency." (Eisenhardt et al. 2010 p. 1266). Collective heuristics also differ from routines in the sense that heuristics are oriented towards a goal and pay attention to the disruptions in an organization's progression towards that goal. Additionally, as Faraj and Xiao (2006 p. 1157) argue, "[r]outines merely emphasize sequences of steps and, thus, are difficult to specify in work situations characterized by novelty, unpredictability, and ever-changing combinations of tasks, actors, and resources." In a recent work on the importance of heuristics to learning and strategy, Bingham and Eisenhardt (2011 p.1457) argue in a similar way that firms learn "heuristics from organizational process experience".

One can also compare collective heuristics with the knowledge integration mechanism "Group Problem-solving and Decision Making" (Grant 1996a). CH:s are strategies or solutions that aim to facilitate knowledge integration, but CH:s do not emerge as an outcome of a certain group process. There was no particular team that was established to create different collective heuristics. Group Problem-solving and other knowledge integration mechanisms are more directly established to solve a particular project task or development undertaking, whereas the CH:s were invented to identify new work approaches and create necessary integrating conditions (Okhuysen and Bechky 2009) when critical problems emerged.

CH:s are not rigid rules or strict directives that steer behaviors (Grant 1996a) exactly. CH:s are broad, leaving room for actual project participants to more carefully specify and create coordination and knowledge integration forms. Finally, a CH may be seen as experience-based trial-and error-method (Lindkvist 2008, Nickersson and Zenger 2004, Thomke and Fujimoto 2000) due to its inherent uncertainty (one never knows whether the strategy will lead one to the goal, or even take one in the right direction) and its intuitive-based character (as opposed to an analytical character).

However, the most prominent characteristic of the concept is not revealed in the comparison with other knowledge integration concepts. The main feature and contribution of the CH concept is the dynamic role that it plays in the knowledge integration process, as explained above. CH:s supported the pacing, timing, synchronization, and orchestration (Söderlund 2010) of different knowledge processes and contributions; *pacing* in the sense of keeping the knowledge integration process ongoing by solving problems and accomplishing a shift in the KIA when needed. The CH:s contributed to the *timing* of the employment of expertise diversity over the course of the project. They fine-tuned the where and when of the use of expertise (Faraj and Sproull 2000), which is something that the KIA could not achieve alone. The CH:s facilitated the *synchronization* of knowledge processes within groups and sub-activities (e.g. "open up for diversity", which involved simultaneous integration of different perspectives, and "bound creativity" which resulted in less time gaps in the design activities). The CH:s also facilitated the synchronization of knowledge processes across the project by quickly enabling new cross-functional interaction forms (e.g. "substantiate upfront" and "cross-check"). The CH:s supported the *orchestration* of the knowledge integration process by complementing the knowledge integration mechanisms and activities in different ways throughout the process (as explained in Chapter 7).

To conclude I have contributed to the field of knowledge integration by answering the call for more process studies that investigate KI agents (Andersson and Berggren 2011) and underlying processes of knowledge integration (Tell 2011). The focus on action or practices, and agents, as also suggested by, for instance Faraj and Xiao (2006), Majchrzak et al. (2012), Lundin and Söderholm (1995) and Engwall and Westling (2004), resulted in improved understanding on how knowledge integration unfolds over time. A model of Dynamic Knowledge Integration was presented, and it was suggesting that knowledge integration contains two interplaying processes; one ongoing process consisting of different knowledge integration mechanisms and activities, and one that is employed when problems that disrupt stability in the process emerge. If problems cannot easily be resolved within existing frames, new strategies, such as collective heuristics, are invented to solve them and to initiate changes in the set of KI mechanisms and activities that are currently in use. This pattern continues until the project reaches its end or until no more critical problems are encountered. However, this is explorative hypothesis generating research and not hypothesis testing research, which means that these relationships need more investigation.

Up to this point, we have answered the research questions, discussed the new model and the importance of focusing on action and knowledge integration agents, and reasoned about the concept ‘collective heuristic’ and the role it plays in knowledge integration processes. Now it remains to suggest potential avenues for future research.

3. Where to next?

I identify and suggest a two-step process for future research. First, since this research is based on but one case study, there is room for more studies on knowledge integration and its underlying processes and activities. The new model on Dynamic Knowledge Integration could be developed and tested in various settings. This could also include further investigation and deeper exploration into the concept “collective heuristic” in order to understand its nature, function, and content better, and to define it more precisely and distinguish it from other well-known KI activities and practices and knowledge integration mechanisms. Future studies also can more exactly investigate the use of collective heuristics in different development contexts and extend the list of different CH:s that are invented and applied in various settings. Taken together, these activities should elaborate on the concept so as to make it more exact and precise, and thereby increase its general value.

When the concept has been further elaborated on, quantitative studies can be conducted to survey and test the extension and use of collective heuristics across different industries, and in different kinds of development processes and projects in order to further examine its value and position relative other KI mechanisms. This can be done to identify different patterns in the use of CH:s. For instance, different types of CH:s might be more common in certain industries than in others, e.g. industries that must develop and launch new products extremely quickly might base its operations to a great extent on this kind of intuitive reasoning, educated guesses or quick and flexible guidelines, whereas in other industries that depend on exactness and precision and involve clear cause-and-effect relationships or high-risk operations may perhaps devalue or forbid this kind of problem-solving strategy. All in all, there should be a great deal of room for more studies on knowledge integration and creation, and its underlying processes, for example, in terms of further investigation of the content as well as the context of the concept “Collective Heuristics”.

REFERENCES

Agar H. M. 1996 *The professional stranger. An informal introduction to Ethnography*. Academic Press

Allen T. 1977 *Managing the flow of technology*. The MIT Press

Alvesson M., Kärreman D. 2011 *Qualitative research and theory development*. SAGE

Andersson H., Berggren C. 2011 Inventors as innovators and knowledge integrators. In *Knowledge integration and innovation. Critical challenges facing international technology-based firms* Eds. Berggren C., Bergek A., Bengtsson L., Hobday M., Söderlund J. Oxford University Press. New York.

Bechky B.A. 2003 Sharing meaning across occupational communities: The transformation of understanding on a production floor. *Organization Science*. Vol. 14. No 3, May-June, pp. 312-330

Berggren C., Bergek A., Bengtsson L., Söderlund J. 2011 Exploring knowledge integration and innovation. In *Knowledge integration and innovation. Critical challenges facing international technology-based firms* Eds. Berggren C., Bergek A., Bengtsson L., Hobday M., Söderlund J. Oxford University Press. New York.

Berkowitz M. W., Gibbs, J. C. 1983 Measuring the development of features of moral discussion. *Merrill-Palmer Quarterly*, 29, 399-410

Bingham C.B., Eisenhardt K. M. 2011 Rational heuristics: the 'simple rules' that strategists learn from process experience. *Strategic Management Journal*, 32: 1437-1464

Brady T., Davies A. 2004 Building project capabilities: from exploratory to exploitative learning. *Organization Studies* (25) 9:1601-1621

Bunderson J.S., Sutcliffe K.M. 2002 Comparing alternative conceptualizations of functional diversity in management teams: Process and performance effects. *Academy Management Journal* 45 (5) 875-893.

Brown J. S., Duguid P. 1991 Organizational learning and communities of practice: toward a unified view of working, learning, and innovation *Organization Science*. Vol. 2. No 1, February.

Brown J. S., Duguid P. 1998 Organizing knowledge. *California Management Review*. Vol. 40 No. 3

Brown S. L., Eisenhardt K.M. 1995 Product development: Past research, present findings, and future directions. *Academy of Management Review*, Vol. 20, No. 2, 243-378

Carlile P. R. 2002 A pragmatic view of knowledge and boundaries: Boundary objects in new product development. *Organization Science*, 13(4) pp. 442-455

Carlile P. R. 2004 Transferring, translating, and transforming: An integrative framework for managing knowledge across boundaries. *Organization Science*. Vol. 15, No. 5, September-October, pp. 555-568

Christensen S., Kreiner K. 1991 *Projektledning. Att leda och lära I en ofullkomlig värld.* Academia Adacta, Lund.

Cohen W.M., Levinthal D.A 1990 Absorptive Capacity: A new perspective on learning and innovation. *Administrative Science Quarterly*, 36 (1990): 128-152

Cook S. D. N., Brown J.S., 1999 Bridging epistemologies: the generative dance between organizational knowledge and organizational knowing. *Organization Science*, Vol. 10, No. 4, July-August, pp. 381-400

Cronin M.A., Weingart L.R. 2007. Representational gaps, information processing, and conflict in functionally diverse teams. *Academy of Management Review*. Vol. 32, No. 3, 761-773

Cross R., Sproull L. 2004 More than an answer: Information relationships for actionable knowledge. *Organization Science*, Vol. 15, No. 4 (July-Aug). pp. 446-462

Daft R. L., & Lengel R. H 1986 Organizational information requirements, media richness and structural design. *Management Science* 32 (5), 554-571

Dahlander L., O'Mahoney S.(2011) Progressing to the center. Coordinating project work. *Organization Science* Vol. 22, No. 4, July-August, pp.961-979

Demsetz H.1991 The theory of the firm revisited. In *The nature of the firm*. Eds. Williamson O.E., Winter S. G. Oxford University Press, New York, pp. 159-178

Dewey J. 2007 (1910) How we think New York: Cosimo

Dougherty D. 1992 Interpretative barriers to successful product innovation in large firms. *Organization Science* 3:179-202

Dunbar K. 1997 How scientists think: On-line creativity and conceptual change in science. In *Conceptual structures and processes: Emergence, discovery, and change.* Eds. Ward T.B., Smith S. M., Vaid J. Washington D.C: American Psychological Association Press

Eisenhardt K. M., Furr N. R., Bingham C. B. 2010 Microfoundations of Performance: Balancing efficiency and flexibility in dynamic environments. *Organization Science* Vol. 21, No. 6, November- December, pp. 1263-1273

Enberg C., Lindkvist L., Tell F. 2006 Exploring the dynamics of knowledge integration. Acting and interacting in project teams. *Management Learning* Vol.37 (2) 143-165

Engwall M. 2002 The futile dream for the perfect goal. In *Beyond project management: New perspectives on the temporary-permanent dilemma* Eds. Sahlin-Andersson K., Söderholm A. Liber. Abstrakt. Copenhagen Business School Press.

Engwall M., Westling G. 2004 Peripety in an R&D drama: Capturing a turnaround in project dynamics. *Organization Studies* 25(9):1557-1578

Faraj S., Sproull L. 2000 Coordinating expertise in softwaredevelopment teams. Management Science. Vol. 46, No. 12, December pp. 1554-1568

Faraj S., Xiao Y. 2006 Coordination in fast-response organization. *Management Science* Vol. 52, No 8, August, pp. 1155-1169

Fiol M.1994 Consensus, diversity and learning in organizations. *Organization Science*, Vol. 5 No. 3 (Aug). pp. 403-420

Galison P. 1999 Trading zone: Coordinating action and belief. In *The Science Studies Reader*. Ed. Biagioli M. Routledge, New York, 137-160

Garsten C. 2004 Etnografi. I *Kunskapande metoder*. Tredje upplagan. Ed. Bengt Gustavsson Studentlitteratur

Gavetti G., Levinthal D. 2000 Looking forward and looking backward: Cognitive and experiential search. *Administrative Science Quarterly*, 45: 113-137

Geertz C. 1973 *The interpretation of cultures*. Basic Books, Inc Publishers. New York

Genzuk M. 2003 A synthesis of ethnographic research. *Occasional Papers Series. Center for multilingual, multicultural research*. Rossier School of Education. Los Angeles

Gersick C. J. G. 1988 Time and transitions in work teams: toward a new model of group development. *Academy of Management Journal*, Vol. 31, No. 1, 9-41

Gersick C. J. G. 1989 Marking time: Predictable transitions in task groups. *Academy of Management Journal*. Vol. 32 No 2, 274-309

Gigerenzer G. 2008 Why heuristics work. *Perspectives on Psychological Science*. 3(1): 20–29.

Gigone D., Hastie R. 1993 The common knowledge effect: Information sharing and group judgment. *Journal of Personality and Social Psychology*. Vol. 65, No. 5, 959-974

Granovetter M. S 1973 The strength of weak ties. *American Journal of Sociology*. 78 (May): 1360-1380

Grant R. M. 1996a Toward a knowledge-based theory of the firm. *Strategic Management Journal*. Vol. 17 Winter Special Issue. 109-122

Grant R. M. 1996b Prospering in dynamically-competetive environments: Organizational Capability as knowledge integration. *Organization Science*. Vol. 7, No. 4, July-August 1996

Hargadon A. B., Bechky B. A. 2006 When collections of creatives become creative collectives: A field study of problem solving at work. *Organization Science* Vol. 17 No. 4 July-August. pp. 484-500

Harvey L. 1997 A discourse on ethnography. In *Information systems and qualitative research*. Eds. Lee A., Liebenau J., DeGross J. I.Chapman and Hall

Hedlund G.1994 A model of knowledge management and the N-form corporation. *Strategic Management Journal* Vol 15, Special Issue: Search for new paradigms (Summer, 1994) pp. 73-90

Von Hippel E., and Tyre M. 1995 How learning by doing is done: Problem identification in novel process equipment. *Research Policy*, Vol. 24, No. 1 (Junuary) pp. 1-12

Hobday M. 2000 The project-based organisation: an ideal form for managing complex products and systems? *Research Policy* 29. 871-893

Huang J. C., Newell S. 2003 Knowledge integration processes and dynamics within the context of cross-functional projects. *International Journal of Project Management* 21 (167-176)

Iansiti M. 1995 Technology integration: Managing technological evolution in a complex environment. *Research Policy* 24. 521-542

Kahneman D. 2011 Thinking, fast and slow. Penguin Books

Kahneman D. 2003 A perspective on judgment and choice. Mapping bounded rationality. *American Psychologist*. Vol. 58, No. 9, 697-720

Kahneman D., Tversky A., 1984 Choices, values and frames. *American Psychologist*, 39, 341-350.

Kameda T., Ohtsubo Y., Takezawa M. 1997 Centrality in sociocognitive networks and social influence: An illustration in a group decision-making context. *Journal of Personality and Social Psychology*. Vol. 73, No. 2, 296-309

Kaplan C. A., Simon H. A 1990 In search of insight. *Cognitive Psychology*, 22, 374-419

Kellogg K. C., Orlikowski W. J., Yates J. 2006 Life in the trading zone: Structuring coordination across boundaries in postbureaucratic organizations. *Organization Science*, vol. 17, No.1, January-February 2006, pp. 22-44

Klahr D., Dunbar K.1988 Dual space search during scientific reasoning. *Cognitive Science* 12, 1-48

Kurtzberg T. R., Amabile T.M. 2000 – 2001 From guilford to creative synergy: Opening the black box of team-level creativity. *Creativity Research Journal* 2000-2001, Vol. 13, Nos 3&4, 285-294

Larson J.R.Jr., Foster-Fishman P.G., Keys C.B. 1994 Discussion of shared and unshared information in decision-making groups. *Journal of Personality and Social Psychology* Vol. 67. No. 3, 446-461

Larson J. R. Jr 2007 Deep diversity and strong synergy: Modeling the impact of variability in Members' problem-solving strategies on problem-solving performance. *Small group research* 38:413

Lave J., Wenger E. 1991 *Situated learning: Legitimate Peripheral Participation* Cambridge University Press

Leonard-Barton D. 1995 *Wellsprings of knowledge: Building and sustaining the sources of innovation.* Harvard business school press

Li H., Bingham J.B., Umphress E. E. 2007 Fairness from the top: Perceived procedural justice and collaborative problem solving in new product development. *Organization Science*. Vol. 18, No. 2, March-April, pp. 200-216

Lindahl M. 2003 *Produktion till varje pris: Om planering och improvisation i anläggningsprojekt*. Doctoral Dissertation. KTH. Fields of Flow

Lindkvist, L. 2004. Governing project-based firms. Promoting market-like processes within hierarchies. *Journal of Management and Governance*. 8(1), 3-25.

Lindkvist, L. 2005. Knowledge Communities and Knowledge Collectivities. A typology of knowledge work in groups". *Journal of Management Studies*, 42(6), 1189-1210.

Lindkvist, L. 2008. Project organization: Exploring its adaptation properties. *International Journal of Project Management*, 26(1), 13-20.

Lindkvist L., Söderlund J., Tell F. 1998 Managing product development projects: On the significance of fountains and deadlines. *Organization Studies*. 19/6 931-951

Lindkvist L., Bengtsson M., Wahlstedt L. 2011 Knowledge integration and creation in projects: Towards a progressive epistemology. In *Knowledge integration and innovation. Critical challenges facing international technology-based firms* Eds. Berggren C., Bergek A., Bengtsson L., Hobday M., Söderlund J. Oxford University Press. New York.

Lundin R. A., Söderholm A. 1995 A theory of the temporary organization. *Scandinavian Journal Management*. No 4, pp. 437-455

Lyles M. A., Mitroff I. I 1980 Organizational Problem Formulation: An empirical study. *Administrative Science Quarterly*. Vol. 25, March

Magnusson T., Lakemond N. 2011 Knowledge integration processes in new product development: On the dynamics of deadlines and architectures. In *Knowledge integration and innovation. Critical challenges facing international technology-based firms* Eds. Berggren C., Bergek A., Bengtsson L., Hobday M., Söderlund J. Oxford University Press. New York.

Majchrzak A., Jarvenpaa S., Hollingshead A.B. 2007 Coordinating expertise among emergent groups responding to disasters. *Organization Science* 18 (1):147-161

Majchrzak A., More P.H.B., Faraj S. 2012 Transcending knowledge differences in cross-functional teams. *Organization Science* Vol. 23, No.4, July-August, pp. 951-970

March J., Simon H.A. 1993 Organizations. Second Edition. Blackwell Publishers. USA

Matusov E. 1996 Intersubjectivity without agreement. *Mind, culture, and activity*, Vol. 3, No. 1

Moeran B. 2007 Ethnography at work Berg Publishers. UK.

Newell A., Simon H.A. 1972 Human problem solving Englewood Cliffs, NJ: Prentice-Hall

Nickersson J.A., Zenger T.R. 2004 A knowledge-based theory of the firm – The problem-solving perspective. *Organization Science* November-December. Vol. 15 No. 6 617-632

Nonaka I. 1994 A dynamic theory of organizational knowledge creation. *Organization Science* Vol. 5, No. 1, Feb.

Okhuysen G.A., Bechky B.A. 2009 Coordination in organizations: An integrative perspective. *Acad. Management Ann*. 3(1) 463-502

Okhuysen G. A., Eisenhardt K. M. 2002 Integrating knowledge in groups: How formal interventions enable flexibility. *Organization Science* Vol. 13, No. 4, July-August, pp. 370-386

Pinto M. B., Pinto J., K. 1990 Project team communication and cross-functional cooperation in new program development. *The journal of product innovation management* 7:200-12

Polanyi K. 1966 *The tacit dimension* London, Routledge. University of Chicago Press.

Polya G. 1945 *How to solve it. A new aspect of mathematical method*. Second edition. Princeton University Press. Princeton, New Jersey.

Prasad P. 1997 Systems of meaning: ethnography as a methodology for the study of information technologies. In *Information systems and qualitative research*. Eds. Lee A., Liebenau J., DeGross J. I. Chapman and Hall

Rennstam J. 2007 *Engineering Work. On peer reviewing as a methid of horizontal control*. Doctoral Dissertation. Lund Business Press.

Rico R.M., Sánchez-Manzanares, Gil F., Gibson C. 2008 Team implicit coordination processes: A team knowledge-based approach. *Academy of Management Review* 33 (1) 163-184

Räisänen C., Linde A.2004 Technologizing discourse to standardize projects in multi-project organizations: hegemony by consensus? *Organization* 11 (1): 101-121

Sapsed & Salter 2004 Postcards from the Edge: Local communities, global programs and boundary objects. *Organization Studies* 25(9): 1515-1534

Sawyer R.K. 2003 *Improvised Dialogues. Emergence and creativity in conversation*. Ablex Publishing. USA

Schmickl C., Kieser A. 2008 How much do specialists have to learn from each other when they jointly develop radical product innovations? *Research Policy* 37 (473-491)

Schön D. 1983 The reflective practitioner. How professionals think in action. Basic Books, Inc.

Sheremata W. A 2000 Centrifugal and centripetal forces in radical new product development under time pressure. *Academy of Management Review*, 25: 389-408

Spradley J. P.1980 Participant observation. Holt, Rinehart and Winston. USA

Stasser G. 1999 The uncertain role of unshared information in collective choice. In *Shared cognition in Organizations: The management of knowledge* Eds. Thompson L.L., Levine J.M., Messick D.M Hillsdale, NJ: Lawrence Earlbaum Associates, Inc.

Stasser G., Stewart D. D., Wittenbaum G. M 1995 Expert roles and information exchange during discussion: The importance of knowing who knows what. *Journal of experimental social psychology* 31, 244-265

Stasser G., Taylor L.A., Hanna C. 1989 Information sampling in structured and unstructured discussions of three- and six-person groups. *Journal of Personality and Social Psychology* Vol.57, No.1, 67-78

Sydow, J., Lindkvist, L. and DeFillippi, R. 2004. Project-based organizations, embedded-ness and repositories of knowledge: Editorial. *Organization Studies,* 25(9), 1475-1489.

Söderlund, J., 2004 Building theories of project management: past research, questions for the future. *International Journal of Project Management* 22 (183-191)

Söderlund J. 2010 Knowledge entrainment and project management: The case of large-scale transformation projects. *International Journal of Project Management* 28. 130-141.

Söderlund J., Tell F. 2011 Knowledge integration in a P-form corporation: Projects epochs in the evolution of ASEA/ABB, 1945-2000 In *Knowledge integration and innovation. Critical challenges facing international technology-based firms*. Eds. Berggren C., Bergek A., Bengtsson L., Hobday M., Söderlund J. Oxford University Press. New York.

Teasley S. D. 1997 Talking about reasoning: How important is the peer in peer collaboration? In *Discourse, tools and reasoning: Essays in situated cognition* Eds. Resnick L. B., Säljö R., Pontecorvo C., Burge B. NATO ASI Series. Series F: Computer and Systems Sciences, Vol 160

Tell F. 2011 Knowledge integration and innovation: A survey of the field. In *Knowledge integration and innovation. Critical challenges facing international technology-based firms*. Eds. Berggren C., Bergek A., Bengtsson L., Hobday M., Söderlund J. Oxford University Press. New York.

Thomke S. H. 1998 Managing experimentation in the design of new products. *Management Science*, Vol. 44, No. 6, June.

Teece D.J., Pisano G., Shuen A. 1997 Dynamic capabilities and strategic management. *Strategic Management Journal*, Vol. 18:7, 509-533

Thomke S. H., Fujimoto T. 2000 The effect of “front-loading” problem-solving on product development performance. *Journal of Product Innovation Management*, 17, pp. 128-142

Tiwana A., McLean E. 2005 Expertise integration and creativity in information systems development. *Journal of Management Information Systems* Summer, Vol. 22, No. 1, pp. 13-43

Tsoukas H. 2009 A dialogical approach to the creation of new knowledge in organizations. *Organization Science*. Vol. 20, No. 6. November-December, pp. 941-957

Tushman M., L. Katz R. 1980 External communication and project performance: An investigation into the role of gatekeepers. *Management Science*. 26(11), 1071-85

Tversky A., Kahneman D. 1974 Judgment under uncertainty: Heuristics and Biases. *Science*, New Series, Vol. 185, No 4157 (Sep. 27, 1974), pp. 1124-1131

Tversky A., Kahneman D. 1986 Rational choice and the framing of decisions. *The journal of business*, Vol. 59, No. 4, Part 2: The behavioural foundations of Economic Theory. (Oct.), pp. S251-S278

Staudenmayer N., Tyre M., Perlow L. 2002 Time to change: Temporal shifts as enablers of organizational change. *Organization Science*. Vol. 13, No. 5, September-October, pp. 583-597

Van de Ven A. H., Delbecq A.L., Koenig R. Jr. 1976 Determinants of coordination modes within organizations. *American Sociological Review*, Vol. 41 (April): 322-338

Van Maanen J. 1988 *Tales of the field. On writing ethnography*. The university of Chicago press

Wegner, D.M., Erber R. P., Raymond 1991, "Transactive Memory in Close Relationships", *Journal of Personality and Social Psychology* 61: 923–929

Weick K., E. 1989 Theory construction as disciplined imagination. *Academy of Management Review*. Vol. 14. No. 4. 516-531

Westling G. 2002 *Balancing innovation and control. The role of face-to-face meetings in complex product development project*. Doctoral Dissertation. Stockholm: Economic Research Institute, Stockholm School of Economics (EFI).

Wheelwright S. C., Clark K. B. 1992 Organizing and Leading "Heavyweight" Development Teams. *California Management Review*. Spring 1992.

Wolcott H.F. 1990 *Writing up qualitative research*. SAGE Publications, Inc.