Competence, methods and practice in information systems development

Omland, Hans Olav

Publication date:
2013

Document Version
Early version, also known as pre-print

Hans Olav Omland

Competence, methods and practice in information systems development

PhD Thesis

Faculty of Engineering and Science
Aalborg University, Denmark

In partial fulfillment of the requirements for the
Ph.D degree in Computer Science

This thesis has been submitted and defended at the
Faculty of Engineering and Science
Aalborg University, Denmark
In particular fulfillment of requirements for the
Ph.D. degree in Computer Science

The defense took place at Agder University on May 27, 2013

Opponents

Ivan Aaen, Associate Professor (Chairman)
Aalborg University, Denmark

Bendik Bygstad, Professor
The Norwegian School of IT, Norway

Jacob Nørbjerg, Associate Professor
Copenhagen Business School, Denmark

Supervisor

Peter Axel Nielsen, Professor
Aalborg University, Denmark

[4] Rejoice in the Lord always. I will
say it again: Rejoice! [5] Let your
gentleness be evident to all. The Lord
is near. [6] Do not be anxious about
anything, but in every situation, by
prayer and petition, with
thanksgiving, present your requests
to God. [7] And the peace of God,
which transcends all understanding,
will guard your hearts and your
minds in Christ Jesus. (Phil 4,4, NIV)

Abstract

Developing information systems appears to be challenging. Systems developers need to enact their competence deploying methods in practice when developing information systems. This thesis seeks to understand and describe the relationships that exist between developers' competence and their deployment of method in practice during information systems development. Methods and practice and the relationships between these are well researched, while less research is focused on competence, its enactment and the relationships between competence, methods and practice. This thesis therefore seeks to answer the following research questions:

Q1. How do competence, methods and practice relate to and influence each other in information systems development?

Q2. How does context influence developers' competence, methods and practice, and the relationships between these, in information systems development?

Answers to the research questions are sought through an explorative research approach. Data were collected through questionnaires, document studies and semi-structures interviews. The answers to the questionnaires were analysed manually. Documents were studied to understand the context of the information systems development and to learn the vocabulary used. The interviews were transcribed, and the transcribed text was sent to the interviewees for verification. The interviews were analysed through several iterations. The unit of analysis was mainly the organisational level, but later the individual level was also included in the analysis.

Answers to the first research question determined that the three elements of competence, methods and practice form intrinsic relationships in information systems development, and that the elements dynamically influence each other through their relationships. The influences may be unidirectional, i.e. one element influences another element directly, bidirectional, i.e. two elements mutually influence each other directly, or indirect, i.e. one element influences another element via the third element. This thesis further finds that it is difficult to separate competence, methods and practice because they are closely intertwined in information systems development.

The second research question examines the influences between the information systems development context and competence, methods and practice. The context of the systems development and competence, methods and practice mutually influence each other in information systems development. The context was further found to influence how communication is performed, how learning takes place and that the staffing of development teams may successfully be based on good relationships between developers and users rather than on formal domain competence.

In summary, this thesis contributes to understand the relationships between competence, methods and practice in information systems development. The intrinsic relationship formed by developer and development influences information systems development. This thesis criticises the strong emphasis on methods and method deployment in information systems development literature and suggests that competence needs to be introduced into the relationships between methods and practice in order to describe what actually happens in information systems development.

Abstrakt

Udvikling af informationssystemer synes at være udfordrende. Systemudviklere må bruge deres kompetence for at anvende metoder i praksis, når de udvikler informationssystemer. Denne afhandling søger at forstå og beskrive de relationer, der eksisterer mellem udviklernes kompetence og deres anvendelse af metode i praksis i systemudvikling. Metoder og praksis og forholdet mellem disse er veldokumenterede i forskning, mens mindre forskning fokuserer på kompetence, dens anvendelse og forholdet mellem kompetencer, metoder og praksis i systemudvikling. Denne afhandling forsøger derfor at besvare følgende forskningsspørgsmål:

Q1. Hvordan forholder og påvirker kompetence, metoder og praksis sig til hinanden i udvikling av informationssystemer?

Q2. Hvordan påvirker konteksten udviklernes kompetence, metoder og praksis og forholdet mellem disse i udvikling av informationssystemer?

Svar på forskningsspørgsmål, søges gjennom en eksperimentel forskningstilgang. Data blev indsamlet ved hjælp af spørgeskemaer, dokument undersøgelser og semi-strukturer interviews. Svarene på spørgeskemaerne blev analyseret manuelt. Dokumenter, blev gjennomgått for at forstå for rammerne for systemudviklingen og for at lære ordforråd som blev brugt. Interviewene blev transskriberet, og den transskriberede tekst blev sendt til interviewpersonerne for verifikasjon. Interviewene blev analyseret i flere iterationer. Analyseenheden var hovedsagelig det organisatoriske niveau, men etterhvert blev det individuelle nivå inkluderet i analysen.

Svar på det første forskningsprojekt spørgsmål fant at de tre elementer af kompetence, metoder og praksis danner tette relationer i informationssystemer udvikling, og at elementer påvirker hinanden dynamisk gennem deres relationer. Påvirkningerne kan være envejs, dvs et element påvirker et andet element direkte, tovejs, dvs at to elementer gensidigt påvirker hinanden direkte, eller indirekte, dvs at et element påvirkninger et andet element via det tredje elementet. Denne tese finder videre, at det er vanskeligt at seperere kompetence, metoder og praksis, fordi de hænger nøje sammen i udvikling av informationssystemer.

Det andet forskningsspørgsmål undersøger påvirkninger mellem systemudviklingens kontekst og kompetence, metoder og praksis. Systemudvikling kontekst og kompetence, metoder og praksis påvirker hinanden gensidigt i udvikling av informationssystemer. Det blev også fundet at konteksten påvirket hvordan kommunikationen foregikk, hvordan læring fandt sted, og at bemandingen af udviklingsteams kan være baseret på gode forbindelser mellem udviklere og brugere snarere end på formel domæne kompetence.

Sammenfattende bidrager denne afhandling at forstå forholdet mellem kompetencer, metoder og praksis i udvikling av informationssystemer. Den tætte sammenhæng som oppstår mellom systemudvikler og systemudvikling påvirkninger udviklingen av informationssystemer. Denne afhandling kritiserer den store vægt forskningslitteraturen innen systemudvikling legger på metoder og anvendelse av metoder og foreslår at kompetence skal indføres i forholdet mellem metoder og praksis med henblik på at beskrive, hvad der faktisk sker i informationssystemer udvikling.

Acknowledgements

I am greatly indebted to the excellence and patience of my excellent supervisor Peter Axel Nielsen, who followed and inspired me throughout the long path to the completion of this thesis. Not only did he open doors for me at Aalborg University, he also challenged me by his thorough and straightforward supervision.

"The first is to become the last" may describe my PhD study. I was the first of 7 PhD students to start studying under a PhD agreement between Aalborg University and the Department of Information Systems at the University of Agder. Five of the PhD students under the agreement have successfully defended their thesis. One student has quit, and I am the last to defend under the agreement since the University of Agder has now established its own PhD program in information systems.

I want to thank all my colleagues at the Department of Computer Science that I met during my stay at Aalborg University in 2001, for their willingness to receive me and for the opportunities to discuss information systems development. Especially I want to thank Ivan Aaen for fulfilling his promise made in 2001 about chairing my PhD committee. I was also granted an opportunity to participate in supervising bachelor students in the study program of the Department of Computer Science at Aalborg University. What I learnt through participation in the supervisor fellowship and through actually supervising student groups was immensely useful for me when returning to the University of Agder. Thanks also go to Lars Mathiassen and Jesper Kjeldskov for their inspiration.

I am indebted to the University of Agder and in particular Carl Erik Moe for his consistent, patient and gentle push to help me finish my PhD, for the flexible working arrangements and financial support during his period as our institute leader during the majority of my PhD work. I am indebted to Tero Päivärinta, Maung K. Sein and Bjørn Erik Munkvold for their useful comments and inspiring cheering along the PhD way. Thanks to Leif Flak, Eli Hustad, Stig Nordheim and Øystein Sæbø for challenging discussions as PhD students reading group. Thanks to the PhD students that have and at present are participating in the PhD program at the University of Agder. It has been a pleasure for me to engage in discussions of different kinds with you all. Furthermore I am indebted to the late Tore U. Ørvik for many enlightening and challenging discussions. Thanks to our present institute leader Bjørn Furuholt, for the final encouragement to finish the summary of this thesis. I am proud of being part of the Department of Information Systems at the University of Agder, and I am thankful for all helpful, enlightening and challenging discussions in which I have been fortunate to be involved.

Thanks also to my co-writers, Tore Junker, Stig Nordheim and Peter Axel Nielsen.

Without the students at the University of Agder, employees of Kristiansand Municipality represented by Arild Sandnes, employees of Visma Unique and an anonymous systems development company I would not have been able to describe the three cases in this thesis.

I further want to thank my dear wife Sigrun and my family for their support and patience throughout these years during which progress on my PhD studies has been by fits and starts.

Contents

Figures

Tables

1. Introduction

Organisations are becoming increasingly complex. Many of them want more extensive use of information systems in order to improve performance when advancing their business opportunities. Hence information systems are made increasingly complex; incorporating more functions than previous systems, offering closer integration within systems and between different systems, and the tailoring of information systems to better fit the organisation's work processes. The continual expansion and improvement of hardware and software technologies provides a basis for the development of even more complex and advanced information systems. However, user organisations and information systems development organisations experience an escalation in information systems projects (Keil et al., 2000) in terms of scope, functionality and price, and find that information systems failures are long-lived (Kautz et al., 2007; Keil et al., 2000; Lucas, 1975; Lyytinen et al., 1987; Sauer, 1993).

1.1 Motivation for the study

Methods have been devised to make information systems development more predictable (Avison and Fitzgerald, 1995; Fitzgerald et al., 2002), to give developers better control over information systems development and thereby make the information system useful to the users. However, an understanding has emerged that methods alone do not give the results expected. Methods are considered unwieldy and inflexible, and are therefore not used in companies (Kiely and Fitzgerald, 2003).

As methods are designed to improve information systems development practice it is reasonable to expect that methods will be easily deployed in practice and that a relationship between the methods and practice is established. However, in a study of the repair of photocopiers the repairmen found that reliance on the repair manuals alone, the so-called canonical practice, was not sufficient to solve some of the problems faced. Something additional was required, namely non-canonical practice (Brown and Duguid, 1991), consisting of stories and anecdotes shared amongst repairmen when describing repair situations. Developing advanced information systems is far more complex than repairing photocopiers. It is therefore reasonable to question the force of the methods alone to succeed in information systems development.

Constructing information systems development methods has been a popular activity for many years. By 1994 already more than 1000 information systems development methods were available (Jayaratna, 1994). Research on both information systems development methods and their deployment is extensive. Methods may be considered explicated practice that can be followed step by step to produce an information system. But resent research finds that information systems development methods are not necessarily understood or deployed by the developers as intended by the method's creators (Fitzgerald et al., 2002; Ørvik et al., 1999; Päivärinta et al., 2010). On the contrary, methods are said to emerge during deployment (Madsen et al., 2006) and are influenced by the developers deploying the methods (Fitzgerald et al., 2002; Madsen et al., 2006; Ørvik et al., 1999). Methods are still considered important as ways of learning new technologies, as a communication tool for exchanging and discussing experiences and for inspirational purposes (Mathiassen et al., 2000).

Developers enact their competence in information systems development with or without deploying methods. Some state that there are differences between competence and the

enactment of competence in practice (Bassellier et al., 2001). Many contextual factors like the developers' personality, motivation, earlier experiences and relationships to other developers may influence how competence is enacted in information systems development practice. This thesis aims to challenge the view that competence can exist without a relationship with practice.

1.2 Problem statement

Method construction and method deployment are greatly studied in the information systems development field. Some research has also been directed at how methods are deployed in practice. Little or no research has specifically targeted the relationships between competence, methods and practice in information systems development. A better understanding of developers enacting their competences while deploying the method in practice may help organise information systems development in a more meaningful way, resulting in better information systems. This motivates a study of the relationships between competence, methods and practice, and the influences between these in actual information systems development. The research focus of this thesis is therefore summarised in two research questions:

Research question 1:

How do competence, methods and practice relate to and influence each other in information systems development?

As the relationships between competence, methods and practice and their characteristics are insufficiently described in the literature on information systems development research, this thesis will firstly seek descriptive answers to research question 1. The relationship between methods and practice has been studied by other researchers, especially regarding how methods are supposed to influence practice. This thesis aims to research and expand the understanding of the relationships and the influences between the three elements of competence, methods and practice in information systems development.

Information systems development always takes place in some context. The context of information systems development may include the organisation in which the system is to be used, the users of the system to be developed, the information systems development organisation and the information systems development team. A study of the relationships between competence, methods and practice alone will not give a full description of what occurs in information systems development and how competence, methods and practice influence, and are influenced by, the context of information systems development. Research question 2 seeks to include this context in the research in order to understand the relationships between competence, methods and practice and the context of information systems development:

Research question 2:

How does context influence the relationships between developers' competence, methods and practice and the relationships between these in information systems development?

1.3 Structure of the thesis

This thesis is based on five publications that contribute in different ways to answer the research questions. In understanding the relationships it is necessary to study both how to acquire competence and methods and to study how competence is enacted and methods deployed in information systems development practice. The learning of competence and methods is studied both theoretically and practically through one case related to students and courses given in a university context. The enactment of competence and deployment of methods in practice are studied through two cases describing information systems development.

In chapter 2, literature describing the theoretical perspectives related to competence, methods and practice is reviewed and descriptions of the relationships between competence, methods and practice and their influences are sought. Chapter 3 describes the research method used in this thesis, while the findings from the research publications are presented in chapter 4. Chapter 5 describes the contributions from this thesis, and chapter 6 concludes the thesis.

2. Theoretical perspectives.

The literature review is divided into two sections. In section 2.1, literature describing competence, methods and practice and the relationships between these in the information systems development and information systems fields is reviewed. In section 2.2 the literature review is expanded to include competence and relationships between competence and practice in general.

2.1 Theoretical perspectives in information systems development and information systems literature

In section 2.1.1 the method used in the literature search is described. Thereafter literature on competence in the information systems development field is reviewed (section 2.1.2) followed by literature on competence in information systems literature (section 2.1.3). Section 2.1.4 details and displays competence, knowledge and skills as reviewed in information systems development and information systems literature. Literature on methods (section 2.1.5) and practice (section 2.1.6) in information systems development is then reviewed followed by a review of literature on relationships between competence, methods and practice in information systems development (section 2.1.7).

2.1.1 Literature review method

Literature for section 2.1 of the literature study was found through searching the major international journals in the information systems field: MISQ, ISJ, ISR and EJIS and other major international journals. The search was made via Ebsco host and other online databases available electronic with search words competence, methods, practice and relationships between competence, methods and practice and the context of information systems development. The search words were combined in different ways. Literature found in the search was used to "Go backward" (Webster and Watson, 2002, p. xvi) to find other literature through searching the references listed in the publications found. Through going backwards literature in other fields than information systems development and information systems was found and were used especially in section 2.2. The "Go forward" step (Webster and Watson, 2002, p. xvi) of the literature review was performed manually by searching the reference lists in the publication found through the "Go backward" step described above.

2.1.2 Competence in information systems development literature

Generally speaking competence in information systems development is described in categories of cognitive competence, functional competence and social competence (LeDeist and Winterton, 2005) even though researchers may use words like analytical skills, development competence or communicational abilities (Lee et al., 1995; White and Leifer, 1986, p. 221).

Systems analysts and designers should be educated as reflective information systems developers being able to demonstrate competence in information systems development by developing computer systems for specific use contexts (Mathiassen and Purao, 2002). However, Mathiassen and Purao (2002) provide no definition of competence or which competencies are needed for information systems developers.

Information systems developers need both technical and behavioural competence (LeDeist and Winterton, 2005; Lee et al., 1995). Competence often includes skills and knowledge which is also the case with Green (1989) whose main concern is how to establish and maintain good relationships between developers and users in information systems development (Green, 1989). Green (1989) operationalizes technical and behavioural skills in an instrument consisting of 21 specific competences to research what competence developers and users need to create and maintain good relationships with users. The 21 competences are a mixture of personal traits (patience, empathy, and listening), technical skills (programming, analysis and design), and training, organisational, and behavioural skills.

Lee et al. (1995) found that information systems development companies were looking for five categories of future employees: 1) programmers for coding, software maintenance, etc., 2) technical specialists, 3) business and/or systems analysts, 4) end user support consultants and 5) computer operators and data entry clerks. The competence needed by the five different categories of employees were divided in four broad categories: 1) technical specialties competence, 2) technology management competence, 3) business functional competence and 4) interpersonal and management competence (Lee et al., 1995). Future employees will in addition need competence to align information systems solutions with business goals and needs, and to build infrastructure for technological integration of information systems. Information systems professionals need to have an in-depth understanding of business functions, technology and change management (Lee et al., 1995). It is therefore strange that Lee et al. (1995) with their wide focus on information systems development professionals use competence only three times in their article and only related to technical competence (p. 331). Neither competence nor concepts of knowledge or skills are defined and skills and knowledge seems to be used interchangeably.

The five most important competencies that are impacting information systems development successes are found in a core team that possesses "Business knowledge (possibly a business systems analyst), good communication skills, technical expertise, analytic skills, and good organizational skills" (White and Leifer, 1986, p. 221). White and Leifer (1986) are in agreement with the other researchers reviewed above but add the word "good" to describe the quality of the competence needed and emphasize that the competencies are on the team level.

The literature review of competence in information systems development renders no single definition of competence. The references to skills and competence in both technical and business areas show that information systems development takes place in many different contexts requiring a wide variety of competences to successfully develop useful information systems for different users in different contexts.

2.1.3 Competence in the information systems literature

There are no clear distinction between competence in the information systems and information systems development literature. In information systems development literature terms like skills, knowledge, and personal traits (Bassellier et al., 2001) are more often used than competence.

Research on competence in information systems covers a wide variety of areas, and views the IT/IS competence of information systems professionals from different perspectives; the concept of IT competence of business managers to champion IT within their organisations (Bassellier et al., 2003), IT competence of business managers to exhibit IT-leadership (Bassellier et al., 2001), organisational information competencies for value creation (Peppard

et al., 2000), critical information systems professional activities and skills/knowledge related to information systems managers (Wu et al., 2007), business competence of information technology professionals (Bassellier et al., 2004), information systems competencies from an information systems functional perspective (Feeny and Willcocks, 1998a), to mention a few.

IT competence is "IT-related explicit and tacit knowledge" (Bassellier et al., 2001, p. 159). The tacit and the explicit forms of knowledge and the concept of knowing are different concepts (Bassellier et al., 2003) where "... knowledge is static and something that we *use* in action, while knowing is dynamic and is *part* of the action" (Bassellier et al., 2003, p. 319). Competence is therefore conceptualised as a duality between what people do and what they possess (Bassellier et al., 2003).

Competence is "the ability to…" perform something (Peppard et al., 2000). However, just defining competence as "the ability to …" raises the question of whether information systems competence is only "the ability to…?" What about the actual performance of competence in information systems? Peppard et al.'s (2000) definition of competence is criticised by Bassellier et al. (2003), who state that competence is grounded in everyday practice. This thesis holds that defining competence only as "the ability to" is too narrow to describe competence in information systems development.

In an IT governance and management framework focusing on information systems capabilities for exploiting information technology, Feeny-Willcocks (1998a) suggest nine core IT capabilities to facilitate the exploitation of IT in a company through three main points: skills, time horizons and motivating values(Feeny and Willcocks 1998). Feeny and Willcocks (1998b) mapped their proposed 9 capabilities (Feeny and Willcocks, 1998a) with the three skill categories of technical, business and interpersonal using the categories low, medium and high to describe the relationships between them. The only occurrence of the category low in the mapping was to describe the relationship between the capability making technology work and the capability business skills. The result seems obvious, as making technology work will be considered as infrastructure, and expected to function without problems by people employing their business skills. Four skills were found critical for high performance in a company: Business Skills Orientation, "Soft Skills", The High Performance Team, and Distinctive Sets of Skills, Attributes and Drivers (Feeny and Willcocks, 1998b).

Willcocks et al. (2006) tested a revised framework of their core information systems capabilities framework (Feeny and Willcocks, 1998a) on three organisations through a longitudinal case study using three perspectives: Business and the information systems vision, Design of IT architecture, and Delivery of IT services. Despite not explicitly testing interpersonal skills, Willcocks et al. (2006) still stated that each of their nine capability was supported by a "distinctive mix of interpersonal, technical and business skills" (p. 29). They also found a recent increase in the requirements "for 'soft' interpersonal skills across all roles, all roles demand high performers, and each role requires a specific set of people behaviours, characteristics and skills" (Willcocks et al., 2006, p. 30). However, Willcocks et al. (2006) did not discuss how competencies related to their suggested capabilities. It is therefore uncertain whether Willcocks et al.'s (2006) competencies and capabilities are synonyms. The uncertainty was not relieved by Willcocks et al. (2006) when describing informed buying and making technology work as general competencies but reverted to using the term capabilities when discussing the findings without giving any explanation of the terms capabilities and competencies.

IT professionals were found to develop "different conceptualizations of competence, varying in scope from narrowly focused on current task requirements to broader requirements of the profession" (Bassellier and Benbasat, 2004, p. 677). Knowledge often used by IT professionals in their work belongs to domains other than IT professionals' technical hardware and software competence (Bassellier and Benbasat, 2004). Reviewing and summarising studies on the competence of IT professionals leads to a proposed taxonomy suggesting "that the business competence of IT professionals comprises knowledge of skill in two broad categories: organization-specific, and interpersonal and management" (Bassellier and Benbasat, 2004, p. 681). Concluding their findings, Bassellier and Benbasat (2004) found that competence has three different aspects: knowledge, skills, and abilities.

Information competence is used "to capture all aspects of the management of information in an organization, including assessing the role of information in an industry, the exploration of the potential impact of technology, the identification of competitive opportunities through to the design of systems, the deployment of information technologies and the realization of business benefits from these systems" (Peppard et al., 2000, p. 293). Peppard et al. (2000) started researching competence in general management literature before turning to information systems literature for specific guidance on information competencies. The information systems literature was found to predominantly focus on individual competence in the form of information systems skill sets, including technical, business and interpersonal skills ((Peppard, Lambert et al. 2000). The implication of a skill set's understanding of competence is that with IT the solution to problems can be solved by equipping information systems specialists with additional skills ((Peppard, Lambert et al. 2000). In their conclusion, Peppard et al. (2000) go on to suggest a framework consisting of three broad competencies: information strategy, IT/IS supply, and information exploitation competencies, that are transformed into six macro competence definitions furthered detailed in a total of 25 micro competencies. This thesis holds that such atomisation of the concept of competence is meaningless in the same way as anticipating that it is possible to solve any problem in information systems just by providing additional skills.

IT literature is found to lack "an in-depth discussion of the specific competence construct and its measures" (Bassellier et al., 2003, p. 318) on how business managers' competence influence the championing of IT, especially on an individual level. Bassellier et al. (2003) contrast knowledge and competence, and state that knowledge is a key part of competence, but that competence is more than knowledge and is grounded in everyday practice. However, IT professionals do not have enough competence to manage IT for a company, and managers of companies need IT competence to be able to perform their work (Bassellier et al., 2001). Competence in IT is not sufficient to explain or predict how individual business managers will perform in IT leadership. A proactive performance using IT depends on factors like motivation, support from superiors, resources available and power in the organisation, in addition to competence in IT (Bassellier et al., 2001).

The only competence construct found in information systems literature that discusses levels of competence was Munro et al.'s (1997) model of competence for end users. The depth dimension rests on the mastery of competence. It seems that most of the competence constructs found in information systems literature seems to assume that competence is enacted at its highest level, without discussing or problematising the actual enactment of competence.

The competence construct (Munro et al., 1997, p. 47) also contains a finesse dimension defined as "the ability to *creatively apply*" end user competence. For some reason Munro et al. (1997) revert to the ability to in their further discussions instead of using the competence of mastery for the depth dimension, which is in agreement with Peppard et al. (2000). Discussing the competence of mastery in work and the ability to perform something through enacting competence places the question on the different levels of competence, or rather whether there are differences in the results of competence enacted.

Competence is learnt. The Dreifus' Model of Skill (Benner, 1984) states that in acquiring and developing a competence the student passes through five levels of proficiency: novice, advanced beginner, competent, proficient, and expert. According to (Brenner 1984), the different levels reflect changes in three general aspects of competent performance, 1) a movement from reliance on abstract principles to use past experiences in practice as paradigms for further actions, 2) a change in the perspective of the situation at hand where focus is directed from details to a more complete whole where only certain parts are relevant, and 3) a passage from observing the situation from the outside to active participation in the situation.

2.1.4 Competence, knowledge and skills

The review of information systems competence reveals no common understanding or definition of competence. The review of competence in information systems development and information systems reveals an overlap between the two.

Concepts of competence, knowledge and skills are often used interchangeably (cf. Table 1) without being defined or contrasted. Table 2 shows very few commonalities in the conceptualisation of competence in information systems competence and information systems development competence. The different authors seem to create their own concepts of competence, except for Bassellier et al. (2001, 2003) and Bassellier and Benbasat (2004) who use the same concept of competence.

References	Competence	Knowledge	Skills
Bassellier et al. (2001)	■	■	■
Bassellier et al. (2003)	■	■	
Bassellier and Benbasat (2004)	■	■	■
Feeny and Willcocks (1998a,b)			■
Green (1989)			■
Lee et al. (1995)	■	■	■
Mathiassen and Purao (2002)	■	■	■
Munro et al. (1997)	■	■	■
Peppard et al. (2000)	■		■
White and Leifer (1986)		■	■
Willcocks and Feeny (2006)	■	■	■

Table 1. Competence in information systems and information systems development contexts

Table 3 presents a closer analysis of the concept of knowledge, revealing some communality between the different authors in the areas of technology, application, systems development, management of IT and access to IT knowledge. Knowledge in the 5 areas mentioned include both information systems and information systems development, as knowledge spans both the knowledge directly related to information systems development and to the domain where the

information systems are to be used. Knowledge within all 5 areas are important for developers that are to develop information systems, since the development must include both the technicalities of developing an information system and making the system useful from a user perspective.

A closer analysis of the concept of skills shows that many researchers found interpersonal communication, business and technical skills important (cf. Table 4). However the categories of interpersonal communication, business skills and technical skills are very wide and generic and do not necessarily mean the same in all publications reviewed.

References	IT comp. of business managers	Ability to deploy knowledge	Knowing	Narrow focus on task	Requirement of profession	Technical	Analysis	Design	Programming	General	Social	User comp. Breath, depth, finesse	Information competence	Macro competence	Micro competences	Client-related core comp.	Core IT competences	Core IS comp. framework
Bassellier et al. (2001)	■																	
Bassellier et at. (2003)	■	■	■															
Bassellier and Benbasat (2004)	■			■	■													
Lee et al. (1995)						■												
Mathiassen and Purao (2002)						■	■	■	■	■	■							
Munro et al. (1997)												■						
Peppard et al. (2000)													■	■	■			
Willcocks and Feeny (2006)																■	■	■

Table 2. Competence concepts

References	Technology	Application	Systems development	Management of IT	Access to IT knowledge	Organisational specific knowledge	Business operations	Management	Interpersonal skills	General knowledge	Hardware	Concepts and practices	Business	Strategic IT service	Tactical IT services	Use of methods	Application domain	Problem solving
Bassellier et al. (2001)	■	■	■	■	■													
Bassellier et at. (2003)	■	■	■	■	■													
Bassellier and Benbasat (2004)	■			■	■	■												
Lee et al. (1995)	■						■	■	■									
Mathiassen and Purao (2002)	■		■			■							■			■	■	■
Munro et al. (1997)		■								■	■	■						
White and Leifer (1986)													■					
Willcocks and Feeny (2006)														■	■			

Table 3. Knowledge concepts

Competence is typically researched and described on an individual level in information systems development. Individual competence is required for creating core competence at the organisational level ((Bassellier, Reich et al. 2001) (Nordhaug 1998). Information systems development competence at the organisational level is important (Peppard et al., 2000). This

thesis is occupied both with the individual level and the organisational level of competence, as the two different levels are expected to influence information systems development in different ways.

References	Personal use of IT	IT projects	Interpersonal comm.	Leadership/management	Knowledge networking	Org. specific knowledge	Business	Technical	Training	Behavioural	Organisational integration	Process reengineering	General skills	Analytical	Organisational	Soft interpersonal	Soft skills	Distinctive sets of skills, attributes and drivers
Bassellier et al. (2001)	■	■																
Bassellier and Benbasat (2004)				■	■	■												
Feeny and Willcocks (1998a)			■				■	■										
Feeny and Willcocks (1998b)							■										■	■
Green (1989)			■			■		■	■	■								
Lee et al. (1995)				■			■	■			■	■						
Mathiassen and Purao (2002)			■											■	■			
Munro et al. (1997)													■					
Peppard et al. (2000)			■				■	■										
White and Leifer (1986)			■					■						■	■			
Willcocks et al. (2006)			■				■	■								■		

Table 4. Skills concepts

Competence narrowly described as skills ties competence closer to a given profession or branch, assuming a predefined task (Bassellier, Reich et al. 2001) to be solved through a fit between the employee's competence and the job's requirements in the predefined task. Solutions to any problem in information systems development may be solved with more competence in the hands of the information systems developers (Peppard et al., 2000). A consequence of the two views mentioned may be that shortcomings in delivering value from information systems may come from a lack of competence among the information systems professionals and developers. This thesis holds that a danger of such view of competence is that systems developers may be misled in how problems can be solved and the problems to be solved.

From Tables 1, 2, 3 and 4 and the above discussion related to the tables it is reasonable to conclude that there is no common understanding and conceptualisation of competence, knowledge and skills in the information systems and information systems development fields. The reason may be that competence, knowledge and skills are researched in many different domains and from many different perspectives, further complicating how to understand and try to establish a generic definition of the concepts.

Peppard et al. (2000), however, are a lone voice, stating that all elements in competence must exist "in a holistic context if a particular competence is to exist" (Peppard et al., 2000, p. 315). As shown in the review so far, most research on competence in information systems and information systems development is trying to describe competence by dividing the competence into many small and narrow competencies, atomising the concept. Competence must be understood and can only exist in a holistic context (Peppard et al., 2000).

This thesis concurs with Peppard et al. (2000) and suggests that competence must be understood in relationship to methods and practice. The literature review shows that literature on competence is very wide and does not converge towards a generic definition. The general concept of competence will therefore be discussed further in section 2.2.

2.1.5 Methods in information systems development

There are more than one thousand different methods (Jayaratna, 1994). The number is possibly much higher today. The terms methods and methodologies are used differently in Europe and the USA. In Europe, methodology "refers to the study of methods" (Iivari et al., 2000, p. 207) while the term in USA means "method" as used in Europe. In this thesis the term method is used in the European understanding.

IS literature defines "method" in different ways. Following are three definitions and a simple comparison of them.

> "An ISD methodology (ISDM) has been interpreted as 'an organized collection of concepts, methods (or techniques), beliefs, values, and normative principles supported by material resources" (Hirschheim, Klein et al. , Iivari, Hirschheim et al. 2000).

> A method may be defined as "an explicit way of structuring one's thinking and actions. Methodologies contain models and reflect particular perspectives of 'reality' based on their embedded philosophical paradigms. A methodology must show 'what' steps to take, 'how' those steps are to be performed and most importantly the reasons 'why' the methodology user must follow those steps and in the suggested order" (Jayaratna, 1994, p. 43).

> A method is "*A coherent and systematic* approach, *based on a* particular philosophy *of systems development, which will guide developers on* what *steps to take,* how *these steps should be performed and* why *these steps are important in the development of an information system"* (Fitzgerald et al., 2002, p. 5, italics by Fitzgerald et al.).

The three definitions include common elements, like the method has an organised, structured, coherent and systematic approach to information systems development. The definitions are further based on some philosophical assumptions which represent the method's particular perspectives. A method contains some models and techniques and exists to guide the developer towards what steps to take, how the steps are taken and why the developer should take the recommended steps in a given order. Definitions made by Jayaratna (1994) and Fitzgerald et al. (2002) have many things in common. However there is an important difference that makes Jayaratna's (1994) definition preferred in this thesis. Methods "contain models and reflect particular perspectives of "reality" based on their embedded philosophical paradigms" (Jayaratna, 1994, p. 43) and include more of the systems development context than Fitzgerald et al.'s (2002) definition stating that methods are "... *based on a* particular philosophy *of systems development*" (p. 5, italics by Fitzgerald et al.).

Methods are based on many implicit and explicit assumptions and views (Iivari and Hirschheim, 1996) including ontological and epistemological assumptions, be they explicit or implicit in the method description. An analysis of 10 Scandinavian information systems development approaches/methods shows that concepts such as scope, value orientation, knowledge interest of information systems development, the role of methods, and the

principle of the process are used in the approaches/methods (Iivari and Lyytinen, 1998, p. 162).

Research identifying and describing methods' features is vast: distinctions between methods (Nielsen, 1989), feature analysis (Iivari, 1994; Olle et al., 1983, 1986), paradigmatic mismatches and drifts in method deployment (Päivärinta et al., 2010), paradigmatic analyses (Iivari, Hirschheim et al. 1998, Russo and Stolterman 2000), processes for requirements engineering (Sommerville and Kotonya, 1998), and combinations of methods ((Vidgen 2002). The pure feature analyses have been criticised for not taking developers' practice and their situational contingencies into consideration when evaluating methods (Nielsen, 1991).

It is argued that the practice of information systems development does not follow methods, and it is a-methodical (Truex, Baskerville et al. 2000). Others follow a similar view and have found through empirical studies that methods emerge through practice (Madsen et al. 2006), that methods are never used by-the-book (Fitzgerald 1997), that methods may be used as rituals to achieve private interests (Robey and Markus, 1984), that it is necessary to have a critical view on the use of methods (Fitzgerald 1996), or that information systems development may be influenced by shifts of the paradigm of the method or by deep-rooted paradigmatic values of the actors in the development (Päivärinta et al., 2010).

A related strand of research on methods has been concerned with tailoring methods to the unique situation in which they are to be used. Tailoring should be based on a soft systems analysis of the development task (Nielsen, 1991). Methods are engineered on the spot to fit the current situation and needs ((Brinkkemper 1996). Reports from empirical studies show how method engineering on the spot has been carried out in practice at Motorola (Fitzgerald et al., 2003) and Intel (Fitzgerald et al., 2006). Developers deploying methods are subject to mismatches and paradigmatic drifts resulting in the version of methods deployed being different from the method described by the method developer (Päivärinta et al., 2010). Developers may use their own "in-house" methods or contextualise methods to specific situations (Kiely and Fitzgerald, 2003).

To further complicate the understanding of a method and its deployment a method may exist in four versions depending on how it is understood and deployed (Ørvik et al., 1999; Päivärinta et al., 2010). The first version of a method formally describes the method, while the other three versions relate to the actual deployment of the method – that is how the developer interprets and understands the method, how the organisation as a whole adopts the method, and how the method is actually deployed in information systems development (Ørvik et al., 1999). Resent research, however, suggests that information systems development actors and developers might benefit more from "tools that help to identify and process the emerging conflicts than tools that aid in developing a technically "perfect" and optimized solution" (Smolander and Rossi, 2008, p. 37). The developers' understanding of emerging conflicts in information systems development will therefore challenge their competence, both for deploying methods in practice and in understanding when and where to deploy the methods in information systems development.

Methods deployments "are supposed to change, and ideally, improve practice. Methods are used because the established work culture does not deliver results in a desirable fashion" (Fitzgerald et al., 2002, p. 9). However, in spite of the increase in designing and developing methods, information systems development projects seem to be difficult to control. Many projects overrun on both cost and time, and deliver less functionality than initially planned. It

is therefore important to research, clarify and describe the tension between formalised methods and "method in action" to achieve a more comprehensive understanding of the tension to deploy methods more efficiently (Fitzgerald et al., 2002). Fitzgerald et al. (2002) describe and suggest "method in action" as a model for information systems development. Researchers criticise the method in action stating that such models build on a static conception of reality ((Madsen, Kautz et al. 2006). The term "emergent method" is therefore suggested and used to describe the dynamics that take place in information systems development ((Madsen, Kautz et al. 2006).

With all the different views on method presented in this section, the evaluation of the methods seems to be challenging. Much research has been directed at evaluating methods (e.g. Jayaratna, 1994; Nielsen, 1991; (Siau and Rossi 1998); Siau et al., 1996, 1997). Evaluation criteria are under-developed (Siau and Tan, 2005). Most research evaluates methods in the context of their use and hence seeks to relate features of methods to particular contingencies, or at least to explain how methods and situations can be matched given a particular situation. In contrast criticism has been directed at the instrumental view of methods, whereby methods are seen as "fitting" particular situations.

Concluding the literature review on methods in information systems development, this thesis maintains that method creators seldom expose their ontological and epistemological assumptions or the possible consequences their assumptions may have for information systems developers that deploy the methods. This thesis further suggests that the lack of exposing the ontological and epistemological assumptions makes deployment of the methods problematic, as developers may deploy methods without considering the consequences in practice of the assumptions made by the method creators. Methods seem to be created based on an atomistic and dualistic understanding of methods which further complicate their deployment in practice. However, in recent research literature on methods of information systems development there is an increasing awareness that learning and deploying methods is a social activity that cannot simply be understood by studying methods and their features. Any deployment of methods in practice will require developers that enact competence in practice. Otherwise machines could deploy methods to develop information systems.

2.1.6 Practice as work

The term practice is used in a variety of contexts with a variety of meanings, like practicing religion, practicing a dance step, practicing law or engaging in business practices meaning to do or perform something habitually or customarily. However, if practice is defined narrowly as to do or perform something habitually, innovation is impossible. Information systems development includes both a systematic predefined part and an artistic part. The artistic part is needed when the developers invent new functions or new ways of creating a system. This thesis therefore challenges a narrow understanding of practice and suggests that practice is what actually happens when developers develop information systems rather than what ought to or should happen according to method descriptions.

Practice in information systems development occurs when developers in a wide understanding of the term work in situations that are complex, uncertain, unstable, and unique (Mathiassen, 1998). Information systems development may be overloaded with value-conflicts. However the value conflicts do not occur before individuals enact their competence from different perspectives, working in teams to create a common understanding of both the task at hand and how to reach the stated goals (Mathiassen, 1998).

This thesis maintains that practice is what actually happens in information systems development. It is therefore meaningless to talk about practice as something that exists as an entity outside the relationships between practice, methods and competence. However, practice is still different from competence and methods as competence and methods can, at least partly, exist as entities in themselves while practice cannot exist outside of the action taking place through the enactment of competence and/or deployment of methods.

2.1.7 Relationships between competence, methods, practice and context in information systems development and in information systems

Methods, competence and practice may form the following three relationships in information systems development: methods and practice, methods and competence, and competence and practice. These three relationships further relate to the context of the actual information systems development.

Table 5 presents a summary of the research literature reviewed on the relationships between methods, competence and practice in information systems development. The table shows the directions of the influences between the elements reviewed. It is worthwhile noting that the relationship between competence and method is the only relationship where the influence seems to go one way, from competence to methods.

	Relationships		
	Methods	*Practice*	*Competence*
Methods	--	(Fitzgerald et al., 2002) (Huisman and Iivari, 2002) (Kiely and Fitzgerald, 2003) (Mathiassen and Purao, 2002) (Ørvik et al., 1999) (Päivärinta et al., 2010)	
Practice	(Fitzgerald et al., 2002)	--	(Fitzgerald et al., 2002) (Mathiassen and Purao, 2002)
Compe-tence	(Brooks, 1987) (Fitzgerald et al., 2002) (Jayaratna, 1994) (Madsen et al., 2006) (Mathiassen and Purao, 2002) (Necco et al., 1987)	(Fitzgerald et al., 2002) (Madsen et al., 2006) (Mathiassen and Purao, 2002) (Westrup, 1996)	--

Table 5. A summary of the reviewed literature on relationships between methods, competence and practice.

In recent years a research interest has emerged regarding the relationships between methods and practice as described in methods in action (Fitzgerald et al., 2002) and the emergent method (Madsen et al., 2006). Such research interest is natural since methods are constructed to inform practice. However, method designers offer little practical advice on deploying methods in practice (Fitzgerald et al., 2002). Many organisations claim that they either do not use any formalised methods or that they use methods developed in-house (Huisman and Iivari, 2002; Kiely and Fitzgerald, 2003). This thesis holds that such a claim is surprising as it is widely believed that system developers' adherence to methods in information systems development will benefit the organisation (Huisman and Iivari, 2002), the information systems' development and the resulting information system. Methods are promoted as

solutions to information systems development problems, but deployment of methods does not necessarily lead to successful information systems. One reason might be that formalised methods seek to avoid a reliance on individual developers' abilities in information systems development (Fitzgerald et al., 2002).

Madsen et al. (2006) view information systems development as a sequence of activities. They argue that their emergent method describes information systems development better than Fitzgerald et al.'s (2002) method in action, because the emergent method is more concerned with what dynamically happens over time in information systems development (Madsen et al., 2006). The actual development process is a result of "a complex web and interplay of enacting and interacting actors and structures" (Madsen et al., 2006, p. 226).

In most research on information systems development, the actors, i.e. people performing the practice, seem to be absent. There are exceptions however. Nielsen (1991) identified different ways in which developers deploy methods in practice depending on their experience and values. The more experienced developers perceived methods as being less necessary. However, the more developers possessed values of the profession, e.g. systematic approach, reflective attitude, education, code of ethics, the more methods were appreciated (though never deployed blindly) (Nielsen, 1989). Developers enact methods in action, with different skill levels and capabilities; they learn over time, have knowledge of the application domain, and have some degree of autonomy, commitment, and exercise personal motivation (Fitzgerald et al., 2002, p. 123–134). Although Fitzgerald et al. (2002) have a concern for all actors' deployment of methods, they limit their research to developers in traditional information systems development and their description of the developers consists mostly of the developers' abilities and is less concerned with the relationships between developers and methods. Päivärinta et al., (2010) use the term method in action to describe different actors in the development, e.g. developers of the system and users of the system, and establish a connection between the actors and the method by discussing the actors' paradigmatic understanding of both the contexts of the development and the method deployed in information systems development.

Information systems developers and other actors often co-develop information systems. It is important to understand all actors' involvement in the deployment of methods and how their involvement influences the deployment of methods. Co-development improves the requirements process and the design process; moreover it furthers organisational implementation, ethical principles, and workplace democracy (Bjerknes and Bratteteig, 1995). This thesis will therefore not only study developers in the traditional understanding of the term, but also include users and other stakeholders generally referred to as developers except where it is important to distinguish between the different categories of developers. Different categories of developers will participate with their unique competence when co-developing information systems. The method experience the developers have combined with their competencies will probably influence how they perceive and deploy methods (Nielsen, 1991, p. 73; Ørvik et al., 1999).

Even if method is one of the information systems development discipline's key features, it is also probably the "true origin of its crisis" (Ciborra, 1998, p. 8) as method only has a value when deployed. Methods and method deployment can act as a social defence, undermine the learning process, and hinder creativity in information systems development (Wastell, 1996). Does information systems development methods really describe what happens in information systems development (Truex et al., 2000)? Developers viewing practice only through the

concepts of methods and the available methodical arsenal will overlook what really happens in practice. Truex et al. (2000) therefore question the privileged view that information systems development "is a managed, controlled process" (p. 60). Is it rather a "Bricolage" (Ciborra, 2002), where developers creatively use whatever is at hand to solve problems at hand?

Walz et al. (1993) observed an information systems development team and were surprised by how difficult it was for the team members to communicate and to achieve a common understanding of the team's tasks during a requirements determination process. There are "discrepancies between the state of the art and the state of practice in using software engineering tools and methods" (Curtis et al., 1988, p. 1268 referring Zalkowitz et al., 1984). Despite many attempts to improve methods and their deployment in practice since Curtis et al. (1988) published their research in the 1980s, problems and practices in information systems development persist (Kautz et al., 2007). It is obvious that the relationships between methods and practice are complicated. Methods are promoted as solutions to information systems development problems, but their deployment does not necessarily lead to effective development or successful information systems.

In the relationship between competence and method the influences from competence to methods have received considerable research interest. Some researchers state that methods are formalised competence. The advantage of such a view is that competence is not needed when deploying the actual methods in practice since the competence is already formalised in the methods. However, to develop information systems is both a technique and an art, or a creative process (Brooks, 1987). Therefore when the developer follows the method strictly, innovation might be precluded (Fitzgerald et al., 2002). To increase the element of innovation it is important that the individual developers enact their competencies (Fitzgerald et al., 2002).

This thesis supports the position that a possible clash between the Weltanschauung (Jayaratna, 1994) of the method creator and the method user will lead to the latter deploying the method in ways that differ from the creator's intentions. The deployment of methods will therefore be influenced by both the developers' competence and their views of software development (Cockburn, 2001). The developer needs to understand the method (Ørvik et al., 1999) before he can deploy it. This thesis maintains that understanding the method can only be achieved by some kind of relationship between competence and method. Mismatches between method and method in use (Päivärinta et al., 2010) indirectly indicate that some relationships exist between methods and practice. However Päivärinta et al. (2010) do not describe the relationships between the developer and method in terms of competence. This thesis concludes that there must be some relationships between the developer and method since the mismatches (Päivärinta et al., 2010) relate to the paradigms held by the developer. As a consequence of the above discussion this thesis concurs with Necco et al. (1987), stating that information systems development's key factors are improved involvement and better personnel including competence; and method in itself does not suffice (Necco et al., 1987). This thesis therefore agrees that "… it is more important to have specialized knowledge about problems and possible solutions than it is to have general knowledge on how to structure and conduct development processes" (Mathiassen and Purao, 2002, p. 83).

Information systems development is always taking place in some context. Contextual factors may be of different types. Contextual arrangements as organisational politics may influence information systems development (Kautz et al., 2007). Other contextual factors that may influence information systems development are users of the system, contracts for the development, the domain where the development takes place, and how the development teams

are staffed. Method creators cannot consider and foresee all these dynamic influences in actual information systems developments. Madsen et al. (2006) focus on the emergent method and state that the developers contribute to the dynamics, leading to the emergent method. However they do not discuss how developers use their understanding of the totality of the information systems development situation in actual development. It is therefore unclear how Madsen et al. (2006) view the relationship between the developer and the development context.

In studying how systems developers work in practice, Westrup (1996) suggests that systems developers' representations of organisations are actively constructed as rational, coherent, and fitting to computerisation. It is therefore important to understand the context of a given information systems development, what preconceptions developers have, how they act and interact with other stakeholders, and how the totality of the situation influences the actual information systems development. To achieve an "understanding" of the totality of the development situation, competence must in some way relate to practice. The reflective practitioner uses competence to reflect on practice, contemplating on both how to proceed in practice and what learning might occur as a result of practical experiences (Mathiassen 1998).

This thesis therefore holds that the relationships between competence, methods and practice need to be extended to include the relationships between these three elements and the context of information systems development, to understand and explain what happens in actual information systems development.

2.2 General theory on competence and practice

The literature review in section 2.1 is expanded in section 2.2 to the area of competence and the relationships between competence and practice in sections 2.2.1 and 2.2.2 respectively. In this part of the literature review the "Go backward" approach (Webster and Watson, 2002) was particularly helpful. Searching the reference list in publications found in information systems development and information systems led to literature within organisational theory.

2.2.1 Competence

In this thesis a more elaborate explanation of competence is needed. Much research in many domains has centred on competence. However it seems that the question "What is competence?" (Hager and Gonczi 1996, Le Deist and Winterton 2005) is still unanswered. The reason may be that the concept of competence seems to differ depending on the contexts in which it is studied and the perspectives chosen for the study. The literature review shows that competence has a wide variety of characteristics (cf. Table 6).

Research on differences and similarities between the understanding of competence in the US, UK, France, Germany and Austria resulted in a general conception of competence in three dimensions: knowledge, skills, and social competences that are necessary for particular occupations ((Le Deist and Winterton 2005). These three dimensions, however, do not provide a full description of competence. Hence a typology consisting of four competences, namely cognitive, functional, social, and meta competence was suggested and categorised along two dimensions, occupational/personal and conceptual/operational (Le Deist and Winterton, 2005, cf. Figure 1), where cognitive competence captures knowledge and understanding, functional competence resembles skills while social competence includes behavioural and attitudinal aspects. Meta competence is a special dimension of competence as "it is concerned with facilitating the acquisition of the other substantive competences" (Le Deist and Winterton 2005). Le Deist and Wintherton's (2005) four-dimensional competence

model may be represented as a tetrahedron, reflecting the unity of competences and the difficulty of separating the four types of competence. The sides of the tetrahedron may represent practical competences combining "elements of the dimensions of competence in varying proportions" (Le Deist and Winterton 2005).

Conceptions/dimensions	**References**
Narrow vs. integrated view	(Hager and Gonczi 1996)
One- vs. multi-dimensional	(Le Deist and Winterton 2005)
Specificity vs. non specificity	(Nordhaug 1998)
Context vs. context-free Macro vs. micro	(Boyatzis 1982, Prahalad and Hamel 1990, Simpson 2002)
Domain vs. generic	(Hager and Gonczi 1996); (Le Deist and Winterton 2005)
Core competence	(Prahalad and Hamel 1990)
Individual vs. organisational	Nordhaug (1998)
Learning vs. unlearning Formal vs. informal	Le Deist and Winterton (2005)
Levels of competence	Benner (1984); Bloom (1956)
Meta competence	Nordhaug (1998), (Le Deist and Winterton 2005)
Cognitive vs. constructive	Garavan and McGuire (2001)

Table 6. Characteristics of competence

Competence in organisations appears to be divided along disciplinary lines ((Lindgren, Henfridsson et al. 2004). Combining the dimension of task specificity with the dimensions of firm specificity and industry specificity gives a framework that has three dimensions: task, industry, and firm specificity with a low and high scale ((Nordhaug 1998); cf. Figure 2). The competence typology suggests six categories of competences from the more generic meta competence in the low task, industry, and firm specificity to idiosyncratic, technological competence in the high "task" and "firm" specificity end.

	Occupational	**Personal**
Conceptual	Cognitive Competence	Meta Competence
Operational	Functional Competence	Social Competence

Figure 1. Typology of competence (Le Deist and Winterton 2005)

Competence in organisations appears to be divided along disciplinary lines (Lindgren, Henfridsson et al. 2004). Combining the dimension of task specificity with the dimensions of firm specificity and industry specificity gives a framework that has three dimensions: task, industry, and firm specificity with a low and high scale (Nordhaug 1998); cf. Figure 2). The competence typology suggests six categories of competences from the more generic meta competence in the low task, industry, and firm specificity to idiosyncratic, technological competence in the high "task" and "firm" specificity end.

Reaching a common understanding of the concept of competence seems to be difficult. The meaning of the same conceptions or dimensions of competence may have different meaning; meta competence is a low level broad competence (Nordhaug, 1998) or meta competence is a type of competence that facilitates "the acquisition of the other substantive competences" (Le Deist and Winterton, 2005, p. 39). Using homonyms to describe different conceptions of competence make it even more difficult to obtain an understanding of the concept of competence.

TASK SPECIFICITY	FIRM SPECIFICITY Low High		
	INDUSTRY SPECIFICITY Low High		
Low	Meta-Competences	General Industry Competences	Intra-Organisational Competences
High	Standard Technical Competences	Technical Trade Competences	Idiosyncratic, Technical Competences

Figure 2. A Competence Typology (Nordhaug 1998)

Competence exists both on a micro and macro level in organisations (Boyatzis 1982, Prahalad and Hamel 1990, Simpson 2002). The micro level relates to the individual knowledge and may partly be viewed as personal traits allowing for "inclusion of the dimensions of behaviour that lie behind competent performance when discussing competence" ((Bassellier, Reich et al. 2001)p. 163). The macro level relates to an organisation's unique combination of core competencies (Simpson, 2002). The two levels of competence, micro and macro, are related in that it is the individual actors in the organisations that perform the tasks needed to get an organisational result (Nordhaug, 1998).

A strand of research on competence concerns the objectives of enacting competence. Competence is used for gaining a competitive advantage ((Prahalad and Hamel 1990). "Core competencies are the collective learning in the organization, especially how to contribute diverse production skills and integrate multiple streams of technologies" (Prahalad and Hamel 1990). Core competence relates to the organisation of work and the delivery of value. However skills and core competencies are not clearly distinguished, as shown in the following sentence: "The skills that together constitute core competencies....." ((Prahalad and Hamel 1990).

This thesis holds that competence has to be researched through a wide approach. Competence cannot be itemised into small atomic parts as an approach to be more competent to develop information systems. A wide approach to competence is important, as a reaction to the criticism that much of the existing view of competence is too atomistic. Concentrating on tasks is the primary reason why so many people lapse into a narrow view of competency standards (Hager and Gonczi 1996). The task view of competence holds that "competency standards are often thought of as simply a series of discrete task descriptions" (Hager and Gonczi 1996). The integrated concept or view of competence is contextualised when developers enact their competence by selecting key tasks or elements that are central to the practice of the profession to which they relate. Such contextualisation requires knowledge about the context where to enact the competence, the tasks to be completed, the result desired from the competence enactment and the assumptions of the methods to be deployed in information systems development.

Researchers of competence seldom explicate philosophical and epistemological tensions related to competence, especially assumptions about the nature of work, the individual and the organisation (Garavan and McGuire 2001). Taylorism (Taylor, 1911)with its one best "way" is the root of the competency movement and the functional view of management. (Garavan and McGuire 2001)Competency models seek at a simplistic level "to identify the ideal combination of skills, knowledge, attitudes and experience, the possession of which enables employees to become high performers with the potential to add value to the organization" ((Garavan and McGuire 2001). Ontologically the functional view makes the developer and the

information systems development into two separate entities treating employees in a rational and quantitative way. Separating developer and information systems development is challenged by developmental humanism positing that employees should be provided with a broad degree of self-control and self-regulation on the basis that this will empower the employees to actively work towards fulfilling the aims of the organisation (Garavan and McGuire 2001).

The competency literature generally espouses a rationalistic, positivistic perspective and makes some important assumptions about work and behaviour (Garavan and McGuire 2001). One important assumption is the strong bias towards considering competency in a context-free way where competencies are atomistic, mechanistic, and bureaucratic, and that the use of such competencies will yield high performance irrespective of the organisational context in which they are used (Garavan and McGuire 2001). Many of these descriptions of competency do not consider the characteristics of the human agent, and offer little consideration of when the competencies are used, how they are used and the moderating influence of personal characteristics on their usage (Sandberg, 2000).

Independent of the different perspectives and characteristics of competence, competence must be learnt. Learning competence happens in the following levels: knowledge, comprehension, application, analysis, synthesis and evaluation ((Bloom 1956). Anderson and Krathwohl (2001) suggested a revision to Bloom's (1956) taxonomy, emphasising two important changes, a) from noun to verb, making it more active and b) making creating the highest level instead of evaluating: resulting in the following levels: remembering, understanding, applying, analysing, evaluating and creating. According to Dreifus' Model (Benner, 1984) a student passes through five levels of proficiency: novice, advanced beginner, competent, proficient, and expert to acquire and develop a competence. According to (Brenner 1984), the different levels reflect changes in three general aspects of skilled performance: 1) a movement from reliance on abstract principles to use past experiences in practice as paradigms for further actions, 2) a change in the perspective of the situation at hand where focus is directed from details to a more complete whole where only certain parts are relevant, and 3) a passage from observing the situation from outside to active participation in the situation. The different levels of competence are seldom used or detailed in research on competence in information systems development. It seems that researchers simply assume that the developers are at the highest level, both in what competence they have and how they enact their competence. This thesis therefore holds that competence, methods and practice in information systems development need to consider the different levels of competence available in a given development.

2.2.1 Competence and practice

Taylor (1911), with his scientific management (early in the 20th century), noted that the most competent workers accomplished their work faster and with better quality than the less competent workers (Sandberg, 2000). Taylor therefore suggested specific training in standardised tasks in order to increase the workers' competence and thereby effectiveness in organisations. A result of Taylor's view of competence may be seen in some information systems research on competence where competence is atomised, i.e. broken down into small pieces that are suggested for solving specific problems. Examples of the atomisation of competence are the conceptualisations proposed by Green (1989), who suggested 21 different competencies for information systems developers to relate to users. Peppard et al. (2000) suggested three broad categories of competence, divided into 6 macro competencies that are furthered divided into 25 micro competences. The micro competencies are then used to solve

specific development problems. The atomisation of competence makes competence generic in that it is fitted to the task without considering the context of the task and the developer that enacts the competence suggested to solve the specific problem. However, the continual development of the understanding of competence has, in recent years, led to criticism of the atomistic and generic conceptualisation of competence (Hager and Gonczi, 1996; Le Deist and Winterton, 2005; Sandberg, 2000).

Sandberg (2000) studied "what constitutes human competence at work" (Sandberg, 2000, p. 9) and found that that the engine optimisers' conceptions of engine optimising influenced their enactment of competence and concluded that "…. worker and work form one entity through the lived experience of work" (Sandberg 2000). Competence is therefore more than sets of attributes "used in accomplishing work" (Sandberg, 2000, p. 20). The rationalistic approach to competence based on a dualistic ontology that divides "…. competence into two separate entities, namely, worker and work" ((Sandberg 2000) p. 11) is strongly criticised since "human competence at work is seen as constituted by a specific set of attributes, such as the knowledge and skills used in performing particular work" (Sandberg 2000).

Identifying competence through job analysis based on scientific principles from rationalistic research tradition result in three attribute-based approaches to competence: the worker-oriented, the work-oriented, and the multimethod-oriented (Sandberg, 2000). The *worker-oriented* approaches focus on the worker and make competence more generic and context-independent. The *work-oriented* approaches try to overcome the criticism of worker-oriented approaches by making work their starting point through identifying central activities needed to perform the actual work. The activities are then transformed into personal attributes needed by the developers fitted to the work at hand. The *multimethod-oriented* approaches also view competence as a specific set of attributes drawing on both the worker-oriented and work-oriented approaches to formulate the multi-method approaches. All three approaches describe competence indirectly, assuming that the worker and world are distinct entities with an "… objective reality independent of and beyond the human mind" (Sandberg, 2000, p. 11). Worker and work are divided into two separate entities where it is possible to describe competence as being independent of the worker, thereby making competence generic and applicable independent of the worker, the context and the tasks to be solved by the worker.

The above criticism is also directed towards the objective epistemology in which work is objective and knowable to a degree after which work is independent of the workers who accomplish it ((Sandberg 2000)). In contrast, workers' experiences of work give meaning to and constitute their competence rather than a specific set of attributes (Sandberg, 2000). The worker and the world are inextricably related, implying that the attributes used by the worker when working are context-dependent or situational where workers will frame their understanding of problem situations through their experience of work. Workers' experiences of work are therefore "more fundamental to their competence than the attributes themselves" (Sandberg, 2000, p. 11). The actual enactment of competence in accompanying work are therefore preceded by and based upon the workers' conceptions of work.

Professional competence is primarily constituted by the professionals' understanding of their work as opposed to knowledge and skills (Sandberg and Pinnington, 2009). When Marx, Wittgenstein and other authors point to practice as the centre of professional competence, the practice-based approaches "*do not really explain how these aspects of professional practice become integrated into and form a specific competence in work performance"* (Sandberg and Pinnington, 2009, p. 1143 italics by the authors). Sandberg and Pinnington (2009) offer an

integrative conceptualisation of competence in work performance using Heidegger's existential ontology suggesting that "work performance can be conceptualized as constituted by three interrelated ontological dimensions, namely human way of being, *others* in human way of being, and *things* in human way of being" (Sandberg and Pinnington, 2009, p. 1144). The human way of being includes the relationship between what the actors are and what they do as distinguished from something they possess. *Others* in human way of being points to the social constituting of professional competence defined by those with whom the actors are engaged in some specific human way of being. The third dimension, *things* in human way of being is important, not in terms of what the things are in themselves but by their usefulness in a particular human way of being. Sandberg and Pinnington (2009) suggest a model for distinct forms of competence in work performance (cf. Figure 3) adding a specific understanding of work to the three dimensions mentioned above.

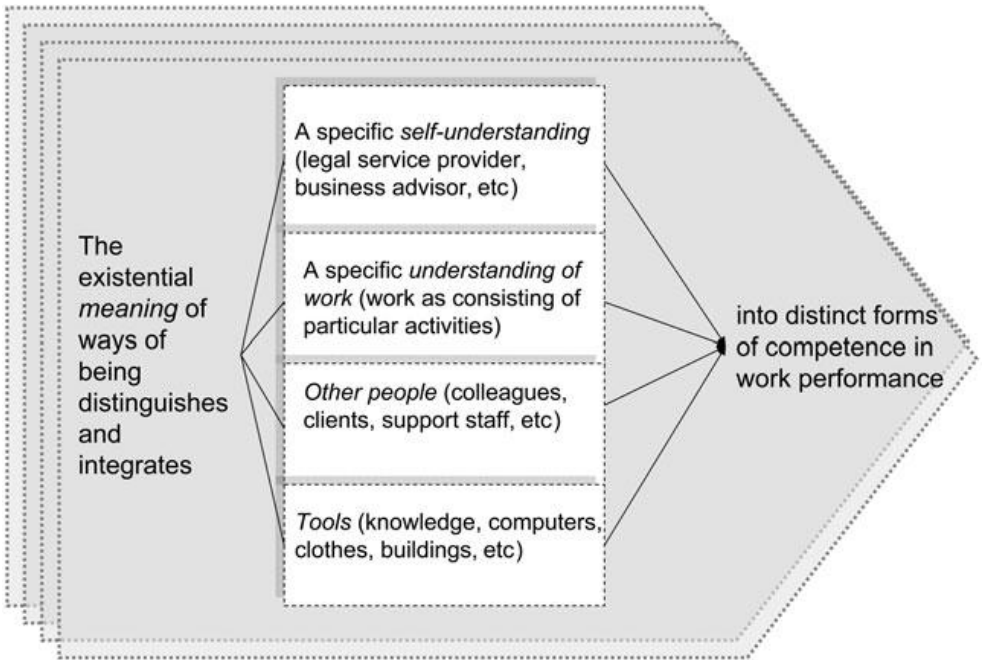

Figure 3. Professional competence as ways of being (Sandberg and Pinnington, 2009)

After reviewing the literature describing many practice-based approaches, Sandberg and Dall'Alba (2009) suggest adopting a life-world perspective viewing practice from a relational perspective where practice is constituted through "the entwinement of life with the world" (p. 1351). The entwinement perspective of practice includes entwinement with others and with things in the world. In that view the individual actor is not the most central, as social order is needed to bring shared meanings to the world in which the actors are living and working. The entwinement of life with world questions the various forms of dualism, such as subject and object, mind and body, thinking and performance that appear in many social theories (Sandberg and Dall'Alba, 2009). The dualism may be overcome by considering relations between, instead of the separateness of mind and body, thinking and performance, etc.

Sandberg and Dall'Alba (2009) use the Heidegger expression "being-in-the-world" to describe how the actors are entwined with the world through engaging in specific tasks. Actors cannot step outside of being-in-the-world. The entwinement with the world is only achieved through the lived body in the actual actions in the world (Sandberg and Dall'Alba, 2009) and resolves the criticism that has been raised against an atomic and instrumental view of competence where the actors enacting their competences are not given full attention. The entwinement between the actor and the world forms the basis for how the relationships between competence, methods and practice are viewed in this thesis. This thesis therefore seeks to introduce the above-mentioned understanding of the relationships between worker and work into the information systems development field.

3. Research approach

This thesis addresses two research questions that seek to understand how competence, methods and practice relate to each other and to the context of information systems development. In chapter 2, existing literature related to the research questions were reviewed. In this chapter the choice of research approach will be described followed by descriptions of the cases studied. Thereafter the processes of data collection and data analysis are described. The chapter concludes with a discussion of validity issues of the research design.

3.1 Choice of case study approach

A case study approach was chosen to answer the research questions in this thesis.

According to Benbasat et al. (1987), questions related to the phenomenon studied can be asked to judge the appropriateness of a case research strategy. The use of case studies is one of several preferred research strategies in explorative research (Yin, 2009). When the phenomenon is in it natural setting with a focus on contemporary events a case research strategy is appropriate (Benbasat et al., 1989; Yin, 2009). In this thesis the phenomenon studied is in its natural setting. The focus of the study is on contemporary events; what actually happens in information systems development and in context. This study does not control or manipulate events in the cases studied (Benbasat et al., 1989; Yin, 2009).

Even if the research questions in this thesis are "how" questions they are actually "what" questions (Yin, 2009) since some "how" questions belong to the exploratory part of the "what" question (Yin, 2009, p. 9; Benbasat et al., 1987). The research questions can therefore be answered by using an exploratory research strategy for which case studies are an appropriate research method seeking to understand the relationships between competence, methods, practice and their context in information systems development and to provide descriptions of these relationships.

A case study research approach can also relate to the established theoretical basis of the phenomenon studied (Benbasat et al., 1987). In this thesis there is an established theoretical basis for parts of the phenomenon studied. The focus of this thesis is to contribute to the established theoretical basis in information systems development as the descriptions in the existing literature of the phenomenon being studied are still insufficient, as shown in the literature review (cf. chapter 2). The information systems development field is a rather young and dynamic field of study, where developers developing information systems are the first to create new knowledge in the field. Researchers can then study how developers work and thereafter develop theories from their studies (Benbasat et al., 1987).

3.2 Case descriptions

This thesis is based on the three cases, 1) the course integration case, 2) the government agency case, and 3) the municipality case. The three cases are presented followed by the rationale for the research design for data collection and data analysis.

3.2.1 The course integration case

The first case study, the course integration case, investigated how study environments provide opportunities for students to increase their competence in developing information systems. In the course integration case students' study environments are approached from two different perspectives, one empirical and one theoretical.

The empirical perspective for studying the students' study environments was an empirical study of how to integrate the more theoretical study environment in the university with a practical study environment in the "real" world. The integration was carried out by establishing a project as an important component in the study environment. The project introduces "real life" or realism into the study environment by letting the student groups practice information systems development in small or medium-sized companies that needed information systems. Each student group had to find their "own" company as part of introducing realism into the practical part of the study environments. When the student groups had successfully found a company that needed an information system, the students in the group had to make the necessary agreements with the company representatives concerning all parts of developing an information system. The students analysed the company's needs for a new information system. Thereafter the students designed, developed and implemented the information system in the company. The information systems development included visiting the company, interviewing the company representatives, participating in steering committee meetings and producing prototypes of the intended system.

In the information systems development the company had to shoulder some responsibility. Each company had to provide a company contact person. The contact person had to be available to the student groups to answer questions related to the information systems development for the customer. If the customer's contact person could not answer the questions raised by the students the person had to find other people at the customer site to answer the questions. The contact person had to participate in three steering committee meetings held during the project.

The students received supervision from the lecturers responsible for the course, and practical assistance from the teaching assistants. The basic assumptions for the empirical perspective of the course integration case were to integrate the theoretical study environment with a practical, realistic study environment. The objectives of the integration of the two study environments were that the students could develop their competence by working in practice in a realistic study environment as described above.

The theoretical perspective in the course integration case was to research how to integrate two courses to provide a study environment for the students to increase their competence in information systems development. The students had to attend the two courses during the same semester in a bachelor program. The courses, a systems analysis and design course and a database course, were integrated through a common project. The idea of using a common project was to let the students experience the results of their own work. In the systems analysis and design course the students performed an analysis of a company's needs for an information system. Based on the analysis the students were then asked to design the system and document the analysis, and the design document was the final product of the systems analysis and design course. The students were then requested to use the document as a basis for constructing a database in the database course.

Part of the theoretical perspective of the course integration case was to study the cost for the lecturers of implementing the integration of the two courses. The result of the theoretical perspective of the course integration case was a theoretical document suggesting how to integrate the two courses. Based on the suggestions in the document an integration of the two courses was successfully implemented in a bachelor program. The actual integration was not part of the course integration study.

3.2.2 The government agency case

The second case study, the government agency case, studied how a systems development company developed a complex information system for a Norwegian government agency. The information system was to be used to keep track of loans, down payments, lack of down-payment, changes in loans, loan guarantees, insurances payments and reimbursements for the insurances payments when down payments are made later than scheduled. The information system contained four interrelated modules, one module for each of the four departments in the government agency. The system was developed by a systems development company that did not have domain competence within the domain areas of the government agency.

The requirements for the system were developed in several steps. The major part of the requirements comprised of using the method of creating user stories. The user stories were produced by the government agency. After the development of the initial system requirements a fixed price contract was signed for the development of the information system. According to the project manager in the systems development company the contract for the development favoured the government agency.

The developers began developing the system by trying to implement the user stories that formed part of the systems requirements made by the government agency. The first efforts attempted to implement parts of the system by implementing some of the user stories. After a period of development the developers discovered that the quality of the user stories was not sufficient to develop the system. The developers therefore started to use the method of prototyping to clarify the requirement specification of the system and thereby improve the quality of the user stories. The use of prototyping proved to be useful and helped the developers in their attempt to develop an information system that was useful for the government agency users.

It was originally estimated that the project would take one year. When the developers were interviewed the information systems development project had lasted for two years with an overrun of 100% on man hours. The information system was not approved by the government agency at the time of the interviews. According to the project manager in the systems development company the information system was close to completion. At the time of the interview 6 developers worked on the development project, down from 15 developers during the most active period of the project.

3.2.3 The municipality case

The third case study, the municipality case, studied a large (by Norwegian standards) municipality that needed to replace its existing information system. The municipality called their information system an Enterprise Resource Planning (ERP) system, since the information system included all the basic functions needed by the municipality. The municipality had to replace their existing ERP system because the provider announced that support for the system was to be discontinued. The municipality searched for a new ERP system that integrated accounting, budget, salary and personnel, invoicing and invoicing module feeding systems, and an e-procurement module. Two systems providers bid for the delivery of the new ERP system. The provider that won the contract was eagerly bidding to win the contract. The systems provider was one of three major systems providers for ERP systems to Norwegian municipalities. The municipality was an advanced user of ERP systems. The winning systems provider wanted to improve its existing ERP system through co-development with the municipality. The municipality was to become the largest

municipality to install and use the vendor's entire ERP system package. The systems provider therefore wanted to take the opportunity to improve their competition advantage over the two other major systems providers for ERP systems for municipalities.

Before the municipality started the tendering process they produced a requirement specification document. The document was produced with input from each department in the municipality that participated in developing the new ERP system. The municipality's project leader and a project consultant used the following method to produce the requirement specification document. The two met separately with each of the municipality's department representatives. From the meetings and discussions with the different department representatives the project consultant aggregated the inputs from each department into a common requirement specification document, including the business processes that should serve the different departments' needs.

When the municipality started their search for a new ERP system their intention was to swap the old ERP system with a new ERP system using the municipality's existing hardware. This intention on the municipality's part influenced the start of the information systems development project. The municipality did not want to invest a lot of resources in the project. After having started the project the municipality changed its mind and participated actively in the development of the new ERP system. The participation included active cooperation with the systems provider. The municipality provided valuable information to the systems provider and co-developed new functions and new systems modules for the systems provider's ERP system. The municipality decided to upgrade their existing hardware and communication facilities to run the new ERP system more efficiently.

The municipality had made a similar change of their ERP system 8–10 years prior to the change of system described in the municipality case. All the central actors from the municipality that took place in the development project researched in this thesis had participated in the former project. The central actors therefore had experience and competence in information systems development related projects. In order to strengthen the municipality's expertise in the project the municipality hired an experienced external consultant to act as a coordinator for the municipality's interests in the project. The result of the cooperation between the municipality, the external hired consultant and the systems provider was two-fold. 1) The developed ERP system was successfully implemented in the municipality. 2) The developed ERP system was a success for the systems provider and gave the systems provider a favourable market position in the municipality market in Norway.

3.3 Data collection

Data from the three cases, the course integration case, the government agency case and the municipality case were collected via questionnaires, document studies and semi-structured interviews

3.3.1 Questionnaires

Questionnaires were used for collecting data in the course integration case. Three different types of actors participated in the course integration case study: students, teaching assistants and representatives for the companies that needed information systems. Four different questionnaires were used, two for the students, one for the teaching assistants and one for the company representatives.
The questionnaires included both closed and open-ended questions (Patton, 1990). Since many of the questions were open-ended the questionnaire gave more qualitative and rich data

than offered by a purely quantitative study (Patton, 1990). Each of the four different questionnaires included several parts in order to collect data from different perspectives. For details of the four questionnaires refer to Appendix B. The main purpose of using questionnaires to collect data was to register the opinions of the different types of actor about the course and the learning consequences that took place for the students that participated in the course.

Over the five month project period (one semester) three steering committee meetings were held. Each student group held its own steering committee meeting aimed specifically at its own project. Participants in the steering committee meetings were students in the student group, the representative of the company related to the student group and the lecturer that acted as supervisor for the student group. Part of the data collection from the students was longitudinal since each student completed the same questionnaire individually after each of the three steering committee meetings held during the 5 month project (course) period. It was therefore possible to register changes that took place for each student between each of the three steering committee meetings. Refer to Appendix B Q1 for a copy of the questionnaire.

At the end of the project period each student group filled in a summative questionnaire about the whole course. Group members from the different groups discussed each question in the questionnaires and chose the group's answers. 15 out of 18 student groups filled in the summative questionnaire. Refer to Appendix B Q2 for a copy of the questionnaire.

Each customer representative was asked to individually complete a questionnaire at the end of the project. The questionnaire was specifically designed for customer representatives, requesting information about the customer representative's view on the work the students did and the course arrangements made by the course lecturers. 15 out of the customers' representatives filled in the questionnaire. For 12 of the student groups both the student group and the customer's representative filled in the summative questionnaires. For the remaining 6 groups either the students or the customers' representatives filled in the summative questionnaires. Refer to Appendix B Q3 for a copy of the questionnaire.

The four teaching assistants that assisted and advised the students in practical matters and systems development processes were also asked to fill in a questionnaire. The questionnaire sought information on the students' experiences during the course and the teaching assistants' contact with the students. The four teaching assistants individually filled in the questionnaire at the end of the course. The questionnaire was designed specifically for the teaching assistants. Refer to Appendix B Q3 for a copy of the questionnaire.

Table 7 gives an overview of the three steering committee meetings held during the project/course, the different questionnaires used and who filled in which questionnaire.

Time line	Activities	Students	Customer representatives	Student assistants
1	Steering committee meeting 1	X	X	X
2	1st time filling in questionnaire 1(Q 1)	X, Q, I		
3	Steering committee meeting 2	X	X	X
4	2nd time filling in questionnaire 1	X, Q, I		
5	Steering committee meeting 3	X	X	X
6	3rd time filling in questionnaire 1	X, Q, I		
7	End of semester summative questionnaires	X, Q 2, G	X, Q 3, I	X, Q 4, I

Table 7. Timeline of steering committee meetings and completed questionnaires (X = participating in the steering committee meetings, Q 1-4 = Questionnaires filled in, I = filled in individually, G = filled in group-wise).

3.3.2 Document

Document data were collected in the course integration case and in the municipality case. All the documents collected in both cases were public documents and were available on paper. Refer to Appendix C for a list of the documents studied.

In the course integration case, documents collected included feedback given by the lecturers and company representatives on the students' behaviour in steering committee meetings. Reports and other documents, written by the students, describing the systems development processes were collected. In addition, course descriptions including learning objectives and the syllabuses for the two courses to be integrated were collected. All the documents were useful for understanding what happened in the course integration case. The documents describing the learning goals and the syllabuses of the two courses to be integrated were central for understanding the relationships between the courses, their objectives and possible points of integration.

In the municipality case the requirement specification document produced by the municipality was reviewed. This document was the only formal requirement specification document produced in the systems development process in the municipality case. As the document was the only document available it was important to study the document as it gave an understanding of the vocabulary used by the municipality. The study of the document also provided information about how the municipality envisioned their future business processes in the different departments participating in the information systems development.

3.3.3 Semi-structured interviews

Semi-structured interviews were used to collect data for the government agency case and the municipality case. An interview guide was formulated, inspired by Patton (1990). Refer to Appendix D for the interview guide. To ensure that the areas to be covered and questions to be asked in the interviews were relevant to answer the research questions in this thesis, the interview guide was tested on an experienced systems developer who was a project manager in a medium-sized information systems development company. Based on the feedback from the test the interview guide was adjusted before it was used in the semi-structured interviews in the government case and the municipality case. The same interview guide was used for data collection in both cases. An overview of all the interviewees in both cases is presented in Table 8.

The objective of this thesis is to study the relationships between competence, methods and practice in the context of actual information systems development. The study of the relationships mentioned was mostly done through the government agency case and the municipality case. Studying the systems requirement specification document made by the municipality was useful for customising the language in the interview to fit the expressions and domain vocabulary used by the municipality. The customisation of the interview guide was made without violating the meaning and content of the questions in the interview guide.

In the government agency case six developers were interviewed (in meetings lasting two hours each). The interviewees included five of the six developers working on the project when the interviews took place and one developer that had been transferred to another project just

prior to the time of the interviews. All the interviewees, including the development company's project manager for the project, were central actors from the development company that participated in the information systems development for the government agency. The interviewees were chosen because they were working on the project at the time of the interviews and had good knowledge about the information systems development for the government agency. The one interviewee that was moved to a new project shortly before the interviews were conducted was proposed by the project manager in the systems development company. The interviews were conducted when the project was in its supposedly final phase of development. The interview data gathered was derived from the interviewees as they retrospectively discussed their experiences in the development process.

In the municipality case fifteen people were interviewed (lasting 1–1.5 hours each). Nine interviewees were from the municipality, including the municipality's project manager, the externally-hired project coordinator, and the sub-project managers for each of the systems modules. All the interviewees from the municipality case were selected as they were central actors in the case. The municipality's project leader was leading the project aided by the externally-hired project consultant. The sub-project managers in the project were all leaders of the departments that were to use the new ERP system.

Six of the interviewees in the municipality case were from the systems development company. They were chosen for the interviews because they were the most central actors from the systems development company that related to and worked together with the actors in the municipality during the project. The interviewees included the information systems development's project manager, the product managers responsible for the different modules of the ERP system, the invoicing system, the e-procurement system, and the domain experts engaged in the project. All the interviews in the municipality case were conducted in retrospect after the main part of the information system has been developed, implemented, and set in production.

The interviews in both case studies were performed in the actual work places of the interviewees, were tape recorded and later transcribed. The transcribed interviews were sent to each of the interviewees respectively for feedback from the interviewees and possible addition of more information. In the government agency no feedback was received. In the municipality case one e-mail was received as feedback on the transcribed interviews. The feedback concerned minor points in the interview.

After the feedback on the transcribed text of the interviews in the municipality case was received two reports were written based on the interview data, the comments from the interviewees and the requirements specification document produced by the municipality. The reports were separately targeted at the systems development company and the municipality, and were sent to the systems development company and the municipality respectively for comments. One reason for making two reports was to ensure confidentiality between what the interviewees belonging to the two different organisations had said in the interviews. Another reason for making two reports was that the different interviewees might be influenced by the other party's interviews when giving their feedback on the information systems development process that was described in the reports. One follow-up interview was conducted with the municipality's project manager to receive comments on the report sent to the municipality. Minor comments were received.

Organisation	Interviewees	Length of interview
Development company in the government agency case	Project manager 5 systems developers	2 hours
Municipality in municipality case	Project manager Project coordinator (externally-hired) Sub project manager salary module Sub project manager e-procurement Sub project manager accounting and "feeding system" Sub project manager web-reporting and invoicing system Sub project manager technical services Sub project manager personnel modules Sub project manager budget module	1–1.5 hours
Development company in municipality case	Project manager ERP-system manager Consultant ERP-system and invoicing system E-procurement manager Support/maintenance responsible ERP- and invoicing module Sales manager at the developer (short interview, was not taped but written from notes and sent for confirmation)	1–1.5 hours

Table 8 Overview of interviews in the government agency case and the municipality case

3.4 Data analysis

Data in this thesis was collected using questionnaires, studying documents and conducting semi-structured interviews. The collected data was analysed depending on which of the three data collection method was used. It is therefore necessary to describe how data collected by different methods were analysed in relation to the data collection methods used. It is further important to describe how the data analysis relates specifically to what was studied.

As theory on information systems development lacks a description of the relationships and influences between competence, methods and practice, this thesis focus both on exploring the relationships and on generating explanatory theory about the relationships and their influences both on each other and in information systems development contexts. The data is therefore analysed with the intention of understanding the relationships and to generate theory (Eisenhardt, 1989) that can be added to existing theory in the information systems development field. "Analyzing data is the heart of building theory from case studies" (Eisenhardt, 1989). However such theory building is difficult. One way to build theory is within-case analysis, where data from one case is analysed. Within-case analysis can be done by documenting the case and using the case documentation for the within-case analysis. Another tactic for theory building from case research is searching for cross-case patterns (Eisenhardt, 1989, p. 540). In this thesis both the with-in case analysis tactic and the cross-case pattern tactic are used for theory building from the documented and studied cases presented.

3.4.1 Analysis of questionnaire data

Questionnaires were used in the course integration case study (cf. section 3.2.1). The questionnaires were analysed both quantitatively and qualitatively as they contained both types of data. The quantitative data were analysed using descriptive statistics (Creswell,

2003). The occurrences of the replies for each question were counted and summarised. All questionnaires were treated in the same way. The analysis resulted in tables of descriptive statistics showing frequencies and means.

The qualitative answers to the open-ended questions were listed in tables and categorised. Answers of similar type were registered and presented. The answers to one of the open questions were categorised in positive and negative comments about the study environment in the course.

3.4.2 Analysis of document data

The documents in the course integration case (cf. section 3.2.1) were read and their meanings were analysed to obtain an overview of the situations described in the documents. Through continual reflection on the different documents, useful information for understanding the case was obtained.

The analysis of the documents describing each of the two courses to be integrated (course integration case) was done by identifying similarities and differences in the course objectives, course plans and other course-related information. The results from this analysis were used to search for ways to integrate differences and similarities between the two courses into a meaningful study environment for the students. The objectives of the integration were to strengthen the students' possibilities for learning competence in a practical and realistic study environment. The result of the analysis was a document suggesting how to integrate the two courses. The document included arguments for the benefits that the suggested integration of the courses would have for the students. One outcome of the analysis of the two courses was a suggestion to establish a common project for the two courses as an important element in the integration of the courses (cf. section 3.2.1).

In the municipality case (cf. section 3.2.2) the requirement specification was read and analysed to get an understanding of the municipality's self-understanding. The requirement specification revealed that the municipality was describing the business processes that the new system should serve. By describing the requirement specifications through business processes the municipality did not dictate what the new system should be, but stated what business processes the new system should serve. Understanding the municipality's intention for the new system was helpful for understanding the information systems development that took place in the municipality case. By producing the requirement specification through business processes the municipality prepared for a more open discussion with the systems development company during the development of the new system.

The requirement specification was further read and analysed to get an understanding of the vocabulary used in the municipality, both domain-wise, application-wise, and in relation to technical issues. Understanding and using the vocabulary of the municipality gave credibility in the interview situations and was helpful in obtaining a better understanding of what the interviewees said in the interviews.

Since the requirement specification document stated the municipality's intentions for the new system, analysing the requirement specification was helpful when using the interview guide. The interviewer could use expressions from the requirement specification that were understood by the municipality's interviewees. The interviews with the developers in the development company became more focused as the developers had studied the requirement

specifications. The developers and the interviewer therefore had a common platform of vocabulary based on the requirement specification document produced by the municipality.

3.4.3 Analysis of interview data

Semi-structured interviews were conducted in the government agency case (cf. section 3.2.2) and the municipality case (cf. section 3.2.3). As this thesis uses an explorative research approach (cf. the beginning of chapter 3) an exploratory approach is used when analysing the interview data. The objectives of the research described in this thesis were to seek to understand the phenomenon studied and describe them to answer the research questions asked. At the same time this thesis also seeks explanatory answers to the research question (cf. section 3.4 second paragraph), to be able to contribute to information systems development theory. Explanatory answers to the research questions build on findings from the explorative research approach used in this thesis.

The analysis of the transcribed interview data was inspired by grounded theory principles (Glaser and Strauss, 1967). Situations described in the interview data was coded in different ways to find categories that explained the situations described in the transcribed text of the interviews. In this thesis the analysis started with the three seed categories or *a priori* codes (Creswell, 2007) of "competence", "methods" and "practice". These three codes/categories represent the main concepts in the research questions. The code "method" was used both for a method and for a part or parts of a method. Even if the analysis of the interview data started with the three seed categories, the transcribed text was read and coded carefully as described in the following steps 1 to 6. The coding was performed with a strategy of being as open and emergent as possible. However, Walsham's (1995) warning against ignoring any existing theory was heeded. The coding, analysis and study of the transcribed text was therefore informed from relevant theory in two different ways. The coding was informed by the literature reviewed and studied as presented in the literature review (cf. chapter 2). The coding was also informed by different ways of ordering the coded data and by using encounters and episodes (Robey and Newman, 1996) when coding the sixth iteration. The same analysis method was used for analysing the semi-structured interviews from both cases.

The interview data in the government agency case was coded as described in iterations 1 to 4. The interview data in the municipality case was coded as described in iterations 1 to 6 below. In all 6 iterations the text was carefully read and coded using Atlas.it.

1. In the first iteration the occurrences of the three seed categories "competence, "method" and "practice" in the transcribed text was coded. For each occurrence of a seed category the transcribed text was carefully marked. An example of the coding may be that an interviewee said that he deployed prototyping to get a clearer understanding of systems requirements. Such text was coded as "method".

2. In the second iteration the transcribed text was carefully read, searching for occurrences of relationships between the seed categories. The relationships sought in the text may be easy to find or they may be stated indirectly by the interviewees. An example of coding in the second iteration may be that one or more interviewees said that they had project competence and that they used a specific method in the project. Based on what they said a code was entered in the text stating that there was a relationship between the categories "competence" and "method". The second iteration of coding resulted in three relationships between the three seed categories: 1) the relationship between "competence" and "methods", 2) the relationship between

"competence' and "practice", and 3) the relationship between "method" and "practice".

3. The relationships found in the second coding provided a starting point for a refined coding in the third iteration, looking for influences in the relationships. The directions of the influences were coded in the transcribed text. The result of this third iteration of coding led to three different findings: 1) the influence between two seed categories were sometimes uni-directional, i.e. going only one way, 2) the influence between two seed categories were sometimes bi-directional, i.e. going from one seed category to another seed category and then back again, 3) in some instances no direct (neither bi-directional nor uni-directional) influence between two seed categories were found. An example of 1) may be that "method" influenced "practice" in a method–practice relationship. However, practice was not found to directly influence method back as would happen in a bi-directional influence. Finding 3) was trickier to discover. In some situations no direct influence between two seed categories were found. The third iteration of coding revealed that in such cases an influence between two seed categories was working via the third seed category. In a reported situation no influence was found from "methods" to "competence". However closer analysis of the transcribed text of the situation revealed that "method" deployed an influence on "practice" in unexpected ways. The unexpected results in the practice of method deployment led the developer in question to reflect upon his experience. The reflection led the developer to reconsider and upgrade his competence to meet similar situations in future information systems development situations. The developers' competence was changed. The change came from an influence that started from method, went via practice to competence that was not registered as a direct influence from the category of "method" to the category of "competence". And since the developer changed his competence he reported that he would deploy the method differently next time.

4. A fourth iteration concentrated on the level of analysis in both cases. The unit of analysis in the government agency case was mainly the individual developer. The individual developer was interviewed to get an understanding of their work situation and how they enacted their competence to deploy methods in actual practical development situations.

   The unit of analysis in the municipality case was the organisational level. Two organisations co-operated to develop an ERP system. During the data analysis for the municipality case it became clear that analysis in the case also must include the individual level to adequately understand and describe the relationships between "competence", "method", and "practice". The actual development is performed by the individual developer even if he co-develops with other developers.

5. After the fourth iteration of coding two reports were made based on the interview data and requirement specification in the municipality case (cf. section 3.2.2). The reports were sent to the development company and the municipality respectively. After receiving feedback on the reports (via e-mail and one follow-up interview with the municipality's project manager) a fifth iteration of coding was performed on the transcribed text of the interviews in the municipality case. In this iteration categories were combined into three topics that described the relationships and influences between "competence", "method", and "practice": namely "Intrinsic Dynamic Relationships", "Common Understanding", and "Organising Vision". The three

categories were overarching categories that related to the categories of "competence", "methods" and "practice". The overarching categories explain parts of the relationships and influences between competence, methods and practice and their relationship to the context of information systems development. The three overarching categories also provide details to the description of the relationships and influences in the relationships between the three seed categories.

6. As the municipality case data was used for a second publication the results of the fifth iteration of coding was also used in a second publication on the municipality case. The following clarifications and changes were made. The findings of the fifth coding were organised into "encounters" and "episodes" (Robey and Newman, 1996), where encounters are concentrated events carrying "opportunities to address prior performance, to express dissatisfaction, and to plan for meeting future needs" (p. 33) and episodes are events of "relatively long periods of equilibrium" (p. 33). The encounters and episodes where then used in the analyses to provide an even deeper understanding of the municipality case. The deeper understanding proved helpful in answering the research questions. The results of the analysis from the sixth iteration was that "competence" and "method" were found to be intertwined to such a degree that it was difficult to say what was "method" and what was "competence" in the actual information systems development. It was further found that in some situations method may be part of the problem in information systems development rather than part of the solution.

3.5 Validity issues

The validity issues discussed in this section relate to the research approach in this thesis. Validity issues related to the results of the research are discussed in chapter 6. This thesis is based primarily on qualitative case studies; a research approach that is suitable for seeking understanding (Braa and Vidgen, 1999; Yin, 2009). The validity issues discussed in this section therefore relates to how the case study approach is designed and used in this thesis.

Validity "...is seen as strength of qualitative research..." (Creswell, 2003, p. 195). According to Creswell (2003, p. 196) triangulation is the most frequently used strategy for securing validity in a qualitative research approach. In this thesis triangulation of data collection is used. As described in section 3.3, questionnaires, document studies and interviews are used for collecting data for documenting the three cases. Questionnaires and document studies were used for collecting data in the course integration case. Interviews were used for collecting data in the government agency case. Interviews and document studies were used for collecting data in the municipality case. Each of the three cases contribute evidence to the same focus of investigation given in the two research questions, namely the relationships and influences between competence, methods and practice and their context in information systems development.

According to Yin (2009, p. 41) using "*multiple sources of evidence*" can secure the quality of research designs in the data collection phase of a research case. This thesis relies on data gathered from three different cases (cf. section 3.3). One of the cases studies students and their study environment. The two other cases study systems developers in information systems development. The information systems development studied in the two cases takes place in two different organisations with different systems developers and systems users. Data for the three cases are gathered from multiple sources of evidence related to types of information systems development and different types of actors studied like students, users of

the information systems and experienced systems developers. The interviewees that were active in the cases represent both the users' side and the development company's side. Multiple evidences were further provided through the unit of study in this thesis. The unit of study included both the individual (the government case) and the corporate level (the municipality case).

The research design satisfies Yin's (2009) multiple case study criteria using several cases to study the same phenomenon. Multiple case study is preferred compared to a single-case study (Yin, 2009). A multiple case design broadens the basis for a more unbiased understanding of the subjects researched. The use of three case studies in this thesis enlarges and enriches the understanding of the three seed categories studied. The three cases, i.e. multiple case study, was designed to target different types of projects and different types of actors with different competences in order to bring about an enlargement and enrichment in the understanding of the seed categories.

Creswell (2003) further suggests "member-checking" (p. 196) as an important strategy for securing validity, including checking the accuracy of the findings by letting the participants in the case study read the final reports and comment on them. Member-checking is in line with Yin's (2009) advice on validity to *"Have key informants review draft case study report"* (p. 41). In this thesis this criteria for validity is fulfilled in two ways. The transcribed interviews from the carefully planned and conducted data collection in the municipality case were sent to the interviewees for review and validation (cf. section 3.3.3 for more details). After receiving minor feedback on the transcribed interviews (municipality case) the text from the interviews were analysed and synthesised into two reports. One report was based on the requirements specification and the interviews conducted in the municipality. The other report was based on the interviews conducted in the development company. The reports were then sent to all the interviewees in the municipality and the development company respectively for "member-checking" (Creswell, 2003) and "draft review" (Yin, 2009). Feedback on the reports was requested. To secure the criteria of member-checking and draft review a follow-up interview was conducted with the municipality's project manager (cf. section 3.3.3 for further details).

Researchers are further advised to use "rich, thick description" (Creswell, 2003, p. 196) to make the setting of the case study visible to the readers of qualitative research findings. In this thesis the descriptions in both the publication and in the findings section (chapter 4) are made so rich and thick that the reader can understand the context of the cases and the background of the findings. The richness and thickness of this thesis gives the reader a further opportunity to read and conclude on the contributions made by this thesis to theory in the information systems development field.

Yin (2009) suggests that to secure validity in a case study one must "*Establish the chain of evidence*" (p. 419). In this thesis such a chain of evidence is visible in the data collection and in the data analysis. The chain of evidence in collecting data is described in section 3.3. In the municipality and government agency cases the interview guide was made from studying and reviewing literature in the information systems development field. The research guide was also tested before it was used. The process of carefully transcribing the interviews is documented in section 3.3. A further element in the chain of evidence was to send the transcribed interviews to the related interviewees for reviewing and validation, ensuring the researcher's understanding of the cases.

The chain of evidence is furthered through the analysis of the requirement specifications of the municipality and the interview data from the government agency case and the municipality case. The established chain of evidence is described in detail through iterations 1 to 6 in section 3.4.3. Through the 6 iterations it is possible to follow the analysis process. It is also possible to verify the iteration process towards the results and contributions to information systems development literature presented in this thesis.

All research using the research method chosen in this thesis relies heavily on data reported by the interviewees in the case situations and the studied documents produced by the developers. The data is then analysed and understood by the researcher. The results of the research may therefore be influences by the researcher's perception and understanding of the collected and analysed data, as is the case in any research of social phenomena. Any generalisation of the findings should therefore be handled with caution.

4. Research publications

Addressing the two research questions stated in chapter 1 resulted in five publications, which are listed in chronological order in Table 9. Since the five publications approach the research questions from different perspectives, the publications will contribute differently and complement each other in answering the research questions. All five publications contribute to answering both research questions. The full text of the five publications can be found in Appendix A.

The five publications will be presented in sections 4.1 to 4.5 and their findings explained. The actual contributions to theory and to practice will be discussed in chapter 5.

Nr	Title	Published
1	Projects as Learning Environments A Case Study in IS Education (Junker, T. and Omland, H.O., 2000)	Sein, M.K et al. (eds): Contemporary Trends in Systems Development. Proceedings of Information Systems Development 2000. Kluwer Academic/Plenum Publishers, pp. 419–432
2	Course Integration as Learning Environment for Increasing Competence (Nordheim, S. and Omland, H.O., 2002)	IS2002 Informing Science + IT Education Conference, June 19–21, 2002 Cork, Ireland
3	Relationships between Developers' Competence, Methods and Practice in Information Systems Development: A Case Study (Omland, H.O., 2004)	Vasilecas, O. et al. (eds): Proceedings of the Thirteenth International Conference of Information Systems Development: Advances in theory, Practice and Education, ISD'2004, Vilnius, Sept. 9–11, 2004, pp. 305–316
4	The relationships between competence, methods, and practice in information systems development (Omland, H.O., 2009)	Scandinavian Journal of Information Systems, 2009, 21(2), pp. 3–26
5	Actors' Competencies or Methods? A Case Study of Successful Information Systems Development (Omland, H.O. and Nielsen, P.A., 2009)	Proceedings of 20th Australasian Conference on Information Systems, Dec. 2–4, 2009, Melbourne, pp. 215–224

Table 9. Research publications

4.1 Publication 1

"Projects as Learning Environments. A Case Study in IS Education."

4.1.1 Presentation

The course integration case. In this publication we studied how students develop their competence through the fourth semester of an undergraduate degree program in information systems. In particular we examined how student's working in groups developed competence when realism was enforced into their study environments by the following 5 elements: Establishing projects, Supervision, Teaching assistants, Steering committee meetings and Reflections, as presented in the following:

- *Establishing projects*
  Each student group had to establish a project and make the necessary arrangements for developing an information system for a company. The arrangements included establishing contact with the company, managing the student group's relationships with the company, negotiating the necessary agreements with the company and

performing the actual information systems development. The company had to appoint a contact person that was responsible for the company's contact with the student group.

- *Supervision*
  Each student group received supervision several times during the project period from the lecturers responsible for the course. Feedback was given on project performance, the information systems development, the product developed and the project report describing what happened in the project.

- *Teaching assistants*
  Each student group had designated teaching assistants. The teaching assistants were students that had previously taken the course and were experienced in working on projects. They were available for the student groups for questions and help in practical matters.

- *Steering committee meetings*
  Each student group were required to prepare and lead three steering committee meetings for the project. The members of the steering committee were the customer's representative, the supervisor for the student group and the teaching assistants for the student group. The steering committee members gave feedback on the progress of the project.

- *Reflections*
  Each student group had to produce a project report. The report was part of their final exam in the course. The students were requested to reflect on their experiences in the project and to describe their experiences and reflections in the project report.

The research method for this publication was a longitudinal study, gathering data through questionnaires. Students were requested to answer the same questionnaire individually three times during the project. Additional data were gathered by requesting the student groups, the customer representatives and the teaching assistants to fill in summative questionnaires at the end of the course (cf. section 3.3.1). All questionnaires included both open-ended and closed questions (cf. section 3.3.1). The questionnaires were analysed using descriptive statistics. In addition, the qualitative answers to the open-ended questions were grouped under the different questions. The answers were then compared with each other. Similarities and differences in the answers were sought to obtain richer data for understanding the different participants' views of and experiences in the project.

4.1.2 Findings

The first finding: The students were found to be very enthusiastic about the enforced realism in their projects. Some of the customers' representatives reported that their student groups worked hard to satisfy their customers. The students reported that they experienced frustrations, conflicts and successes in their information systems development work. However, from their answers in the questionnaire it seems reasonable to conclude that the students' experiences increased their willingness to experiment with enacting competence and deploying method in practice to solve problems that were not easily anticipated. The students aired a paradox in their willingness to experiment and learn when they complained about the workload in the course, stating that the project created too much work for them, and at the same time they expressed their desire for more projects with enforced realism. It is therefore

reasonable to anticipate that the study environment used in the course increased the students' learning. However, based on the data analysis it was not possible to establish definite causalities between the realistic project and increase in the students' competence.

The second finding: Analysing the data gathered from the students on enforcing realism into the students' projects shows that competence, methods and practice are related in some ways in the information systems development that took place in the projects. But the data did not render information on how competence, methods and practice relate to each other.

The third finding: It seems that students' understanding of the relationships between competence, methods and practice developed over time and depended on their experiences in the course. Data from the steering committee meetings showed that many student groups obtained a better understanding of the task at hand and delivered improved versions of their information systems as their projects progressed. Therefore the study shows that the relationships between competence, methods and practice must be understood to evolve over time.

4.2 Publication 2

"Course Integration as Learning Environment for Increasing Competence."

4.2.1 Presentation

The course integration case. The research leading to Publication 2 was influenced by findings published in Publication 1. Publication 1 studied a course taken in the fourth semester of an undergraduate study. Was it possible to obtain the same learning effects as found in Publication 1 by integrating two courses in the third semester of a bachelor study program? Publication 2 studied, theoretically, the establishment of a study environment that integrated two related courses to be taken in the third semester of an undergraduate study program. In particular, Publication 2 studied the effects that the proposed study environment could have on the information systems development competence of the students. The two authors of Publication 2 were the lecturers responsible for the two courses to be integrated.

The central idea in proposing to establish the specific study environment was to give the students an opportunity to learn from their own experiences in information systems development. The learning effect could be reached through establishing a common project for two courses, a systems analysis and design course and a database course. The student groups taking the two courses were to develop an information system for a university. In the systems analysis and design course the students were to be asked to produce an analysis and design document for the proposed information system based on interviews to be conducted with the users of the system. The student groups were then to produce an analysis and design document in the systems analysis and design course. The produced document was then to be the basis for constructing a database in the database course. Having to use their own analysis and design document to implement a database was anticipated to give the students opportunities to experience the consequences of their own analysis and design work when trying to construct the database. The quality of the analysis and design document was expected to influence the database construction activities.

After Publication 2 was published the proposed integration of the two courses was successfully implemented in the third semester of a bachelor study program. Later a third course, a programming course, was also integrated with the systems analysis and design course and the database course. The systems design made in the systems analysis and design

course and the database constructed in the database course were coded in the programming course. All three courses were to be taken in the third semester of a bachelor study program, and one common project was to be used as basis for the integration between the three courses.

The research method for this publication was to study and analyse documents related to the two courses to be integrated (cf. sections 3.3.2 and 3.4.2). The study and analysis of the documents revealed similarities and differences in course objectives and schedules for the two courses. Through the analysis of the course objectives and schedules for the two courses a common understanding of the uniqueness and similarities of the courses were reached. The understanding reached was used when constructing the proposal for integrating the two courses.

Through the research for Publication 2 the researchers were inspired by and used a pragmatic unpublished competence model developed by a colleague at the University of Agder and by Harden's (2000) integration ladder to enlighten us about curriculum planning for designing study environments.

A full description of the pragmatic unpublished competence model and of Harden's (2000) integration ladder is presented in Publication 2. Since the pragmatic unpublished model is not a formal model of competence it was not presented in the literature review (chapter 2) of this thesis. The model suggests a hierarchy between five competencies where professional/technical competence, action competence, social competence and learning/change competence together form the competence of mastering. The competence of mastering in the information systems development context means that the developer is able to master and actually perform the tasks at hand in information systems development using the four different competences: professional/technical competence, action competence, social competence and learning/change competence. The pragmatic unpublished model has served as an inspiration for furthering the research of the relationships between competence, methods and practice.

The integration ladder (Harden, 2000) presented in Publication 2 is a tool for curriculum planning and evaluation. The model suggests 11 levels of integration between activities or courses or disciplines in learning environments. Levels 1–4 emphasise disciplines while levels 5 to 11 emphasise integrated teaching and learning across different disciplines.

4.2.2 Findings

The first finding: From a thorough document study where the researchers analysed the documents describing the two courses both individually and by comparing them and the researchers' own knowledge of the courses the researchers succeeded in suggesting an integration of the two courses. The researchers suggested that it was possible and meaningful to develop a study environment through integrating the two courses (cf. section 4.2.1). Based on a careful document study, the findings in Publication 1, informed by the theory presented in Publication 2, determined that students can develop their competence in information systems development through the suggested study environment. The researchers' suggestions are based on the provision of opportunities for the students to experience the consequences of their analysis and design of information systems when they create a database for an information system.

The second finding: It was suggested that using a common project for integrating two courses was important for two reasons: 1) the students may experience the consequences of their own

work and 2) the motivational factor that such type of integrated study environment may have on the students. These two points are further explained in the following. For point 1) it was anticipated that the quality of the analysis and design document produced in the systems analysis and design course had a direct influence on constructing the database in the database course. For point 2 it was expected that the experiences the students were expected to obtain when they had to use their own analysis and design documents to construct a database would become the raw material for their reflections on their own performance in the analysis and design course. Through the students' reflections they would have opportunity to change their ways of working and thereby increase their competence in developing information systems, and consequently increase their motivation for developing information systems and increase their competence.

The third finding: Through the analyses of the course objectives and informed by theory it was found that learning may occur on one level of the integration ladder (Harden, 2000) for the students, while the integration of the courses may occur on another level by the lecturers responsible for the courses (cf. section 4.4.2 and Publication 2). Students may experience an integration on level 8 while the lecturers only need to integrate the courses on level 2 in Harden's (2000) ladder. As the cost for lecturers of integrating courses is expected to increase with a higher level of integration it was anticipated that the integration seems to be less costly for lecturers when they integrate the courses on level 2, and at the same time offers the students the benefits of a study environment that integrates the courses at a higher level on the integration ladder. The higher level of integration offers a greater challenge, and at the same time a more fruitful study environment for the students. Given these circumstances the learning that takes place for the students will to a large degree depend on how active they are and how they participate in the learning process.

Publications 1 and 2 were written before the formal PhD thesis work started. Some of the findings in Publications 1 and 2 initiated the search for the relationships between competence, methods and practice. Publications 1 and 2 are interpreted in this thesis with hindsight through the perspective of the relationships between competence, methods and practice.

4.3 Publication 3

"Relationships between Developers' Competence, Methods and Practice in Information Systems Development: A Case Study."

4.3.1 Presentation

The government agency case. Publication 3 describes a case study researching the relationships between competence, methods and practice in an information systems development. The study was conducted in a systems development company. The company developed a complicated information system for a government agency. The information system was to handle and keep track of financial transactions and engagements over time (30 years or more), including the registration of payments and repayments of loans. The system should be able to register, contain and compute information in different currencies. The government agency operated in a very specialised domain and was the only agency of its kind in Norway.

Initially the development company did not have domain competence in the government agency's domain. The developers reported that the four different departments in the government agency did not have a common understanding of concepts used by the agency. The lack of common understanding created problems for the developers during the

information systems development as the developers carried out the initial phase of the development.

To save time and money it was agreed that the government agency should make the use cases for the information system without assistance from the development company. This was not a good idea. After a period of development of the information system the developers found that the use cases were not sufficiently specific. The government agency and the developers then co-developed new use cases. However, even the co-developed use cases did not describe or clarify all the requirements for the new system. At that point in the information systems development an important decision was made. The decision became central for the findings in the government agency case. The developers decided to deploy the method of prototyping for further understanding the needs of the government agency.

The research method in the government agency case study was interviews of all 5 developers that were active in the information systems development when the interviews were conducted, and one developer that had left the project team to perform other tasks just prior to the interview being conducted. The data collection was done through 2 hours of semi-structured interviews with each of the developers at their workplaces (cf. section 3.3.3). The interviews were carefully transcribed by the interviewer. The transcribed text was then coded using Atlas.it. The analysis was inspired by grounded theory (cf. section 3.4.3). The data was coded in four iterations. The first iteration registered occurrences of the three seed categories of competence, methods and practice. The following two iterations aimed to understand the relationships between the three seed categories, while the fourth iteration concerned the level of analysis of the data.

4.3.2 Findings

The first finding: Analysis of the data reveals that there was a decoupling between competence in "how to" prototype and competence in "when and where" to prototype. The developers did have competence in the technicalities of how to prototype, but did not have competence in when and where to deploy prototyping in actual information systems development. The developers reported that they deployed prototyping for gathering and clarifying requirements. However, prototyping was used rather late in the requirement gathering phase. The developers, in retrospect, said that they should have started to prototype earlier.

The second finding: The developers found that the different government agency departments' use of concepts was not consistent between the four departments that were to use the new system. Employees from the different departments often used the same concepts but with an inconsistency in meaning. The employees in the different departments were not aware of these differences in meanings of the same concepts used by the different departments. The developers did not discover the inconsistencies in meaning when analysing the situation. There were no elements in the analysis method deployed that helped them figure out these differences in meaning. However, when the developers had started the development and needed to integrate the modules to be used by different departments into one consistent information system the developers discovered differences in meanings in the concepts used by the different departments.

The third finding: General communication competence was not sufficient to get a common understanding of the requirements of the information system to be developed in the government agency case. Communication needed to be domain specific. Domain specific

communication proved to be a challenge since the government agency was working in a very specialised domain and the developers did not have such domain competence. Through working together with the government agency's employees the developers slowly learnt more about the government agency's domain and improved their competence in the domain. Increase in domain competence helped the communication between the developers and the government agency, and was necessary to clarify the needs of the agency before the developers could develop a useful information system for the agency.

The fourth finding: Three central documents were used in the information systems development: the development contract, the use case descriptions and the design document. The documents were written to clarify the development goals, conditions and context. But the lack of consistency and coherence between the three documents turned out to be a problem for the developers. The three documents were written with different objectives. The development contract may be used as an example. It gave the government agency great freedom to change their mind about the information system to be developed, more or less at any time during the development process. Since the government agency changed their mind several times during the information systems development the developers had to re-develop parts of the already developed information system for it to be accepted by the agency.

4.4 Publication 4

"The relationships between competence, methods, and practice in information systems development."

4.4.1 Presentation

The municipality case. The case describes how a municipality had a new information system developed (cf. section 3.2.3). The development company that developed the information system was a specialised information systems development company and a large actor in the Norwegian municipality market. The systems development company wanted to upgrade, improve and expand its information systems aimed at the Norwegian municipality market. Since the municipality had some of the best domain expertise in Norway in some domains, which were important to the development company, the development company wanted to cooperate with the municipality in upgrading and expanding their information system aimed at the municipality market. The developers and other actors in the development company was greatly motivated by this opportunity and worked hard, both to win the contract for the information systems development project and to make the developed information system a success.

The cooperation between the municipality and the development company was relatively heated at times and demanding discussions took place. However, in the end the developed information system turned out to be successful. For the municipality the success was that they got a new system with some new features, especially within e-commerce and a special invoicing system for billing garbage collection, water consumption and taxes on housing. The system was successful for the development company in that they won a larger share of the Norwegian market, partly due to the improved and expanded information system developed in cooperation with the municipality.

Data collection for this publication was done through document study of a systems requirement document (cf. section 3.3.2) and through semi-structured interviews (cf. section 3.3.3). The document study was used to understand and analyse parts of the municipality's requirements for the new information system and to learn the vocabulary used by the

municipality. The semi-structured interviews included all the main actors from the municipality and the information systems development company that participated in the information systems development. The interviews were carefully transcribed and then sent to the interviewees for verification. Two reports were made based on the interviews and sent to the interviewees for verification.

The data analysis was inspired by grounded theory, starting with three a priori categories of "competence", "methods" and "practice". The transcribed semi-structured interviews were analysed through five iterations (cf. section 3.4.3). The first iteration registered the occurrences of the three seed categories. The following two iterations were aimed at understanding the relationships between the three seed categories, while the fourth iteration concerned the level of analysis of the data. In the fifth iteration categories were combined into three topics that described the relationships and influences between the three seed categories.

4.4.2 Findings

The first finding: Changes in one of the elements of "competence", "methods" or "practice" led to changes in one or both of the other elements. An example of such a change was that changes in method deployment led to changes in practice. Another example was that one developer deployed a method and experienced results in practice that he did not expect. He said that next time he would deploy the method differently because of the unexpected results his method deployment had on practice. Therefore, some kind of relationships exists between the three elements in information systems development.

The second finding: It was found that the changes in and between the three elements mentioned in the first finding happens to all the three elements of competence, methods and practice, and to the three relationship between them, namely the relationship competence-methods, the relationship methods-practice and the relationship practice-competence.

A further analysis of the second finding found that the influences between the elements of competence, methods and practice in the three relationships mentioned were different. Sometimes the influences were found to be uni-directional, meaning that one element in the relationship directly influenced the other element, exemplified by method deployment influencing practice. Sometimes the influences were found to be bi-directional. One element in the relationship influenced the other element in the relationship. Then the other element, sometimes instantly and sometimes later, influenced the first element, such as in the way method influenced practice and practice in turn influenced method. Sometimes the influence of one element on another was found to go via the third of the three elements. One example of such an influence was that the influence from method to competence travelled via practice. One developer said that he experienced unexpected results of his method deployment in practice. His reflections on the unexpected result led to changes in his competence. The developer said that next time he deployed method in practice he would deploy the method differently.

The third finding: Summarising findings one and two presented above shows that relationships exist between competence, methods and practice in information systems development. Finding one states that changes in one of the elements of competence, methods or practice influences another of the elements. Finding two states that the influences between the three elements in information systems development apply to all the three elements, between the three elements and that there are different types of influences between the three elements.

A further consequence of findings one and two and the above argument for finding three means that the relationships between the elements of competence, methods and practice are dynamic, since changes in the any of the three elements led to changes in one or both of the other elements.

The fourth finding: A consequence of findings one to three is that competence, methods and practice form intrinsic relationships. The most important argument for the fourth finding is the third type of influence between the elements presented in finding two. It was found that sometimes the influences from one element to another element went via the third element. The three elements belong together and they form intrinsic relationships in information systems development. It is not possible to explain influences between competence, methods and practice in information systems development if one of the three elements of competence, methods and practice is omitted in the explanation.

The fifth finding: A good person to person relationship, based on earlier cooperation in information systems development between a developer in the development company and a sub project manager in the municipality, was regarded as more important than specific domain competence when staffing a team to develop some specific module of the information system. The two were chosen to develop a specific module together even if they did not have domain competence in the actual domain. Both of them increased their domain competence when they successfully co-developed the actual systems module.

4.5 Publication 5

"Actors' Competencies or Methods? A Case Study of Successful Information Systems Development."

4.5.1 Presentation

The municipality study. Publication 5 continues to explore the relationships between competence, methods and practice, focusing on the relationships between competence and methods. The case background for the municipality case was presented in section 4.4.1 and is therefore not repeated here. Publication 5 studied the effect that method deployment has on information systems development. Specifically the question of whether method deployment created more problems than solutions or help in information systems development was studied.

The research method used was the analysis of the requirement specification document and analysis of semi-structured interviews using the same data as Publication 4. The research builds on the analysis done for Publication 4 (cf. sections 3.4.3 points 1 to 5). The findings from the fifth coding of Publication 4 (cf. section 3.4.3 point 5) were coded in an extra iteration for Publication 5. The coding in the extra iteration organised the data into "encounters" and "episodes" to analyse, explain and present the findings (cf. section 3.4.3 point 6).

4.5.2 Findings

The first finding: The analysis of the municipality case data shows that competence and methods was intertwined in this information systems development. It was not possible to distinguish what was competence and what was method in the actual information systems development practice.

The second finding: In the municipality case meta competence was used when manning a development team for parts of the development. Good prior relationships between a developer and a sub project manager were deemed to be more important than domain competence within the domain area of the information systems development. One major activity in designing and carrying out information systems development is to consider how the methods to be deployed and the developers' competence present in the development team can make a good dynamic fit from start to finish of the information systems development. One way of doing this is through the enactment of meta competence. Meta competence facilitates the acquisition of the other competencies and their enactment (cf. section 2.1.1 and Figure 1).

The third finding: Publication 5 suggests that one way of broadening the understanding of competence is to view the enactment of competence and the deployment of methods as integrated and mutually influencing each other in information systems development. Without an understanding of how competence and method are intertwined it is easy to focus too much on the deployment of methods. Research results presented in Publication 5 therefore suggest that too much focus on methods in information systems development can lead to a less successful development. The reason for the suggestion is that the developer joins an information systems development project as a person bringing the totality of him/herself, not just his/her competence, into the information systems development. Therefore, deploying methods in practice without considering how and when a method is deployed and how competence is enacted in practice may lead to method deployment being counterproductive in information systems development.

5. Contributions

This chapter discusses contributions based on the findings presented in chapter 4. Presentation of the contributions is related to the two research questions that this thesis seeks to answer. The two research questions and a general overview of the contributions are presented in Table 10, followed by a detailed discussion of the contributions related to each research question.

Research questions	Contributions
Q1. How do competence, methods and practice relate to and influence each other in information systems development?	The overall description of the contribution: A description of influences and relationships between competence, methods and practice in information systems development. Contributions to answering research question 1 are presented in the following sections: 1. Adding competence to methods and practice in information systems development 2. Influences exist between competence, methods and practice 3. Competence, method, practice and the relationships between them are dynamic during information systems development 4. Competence, methods and practice form intrinsic relationships 5. Competence and method are intertwined 6. Developer and information systems development are intertwined
Q2. How does context influence the relationships between developers' competence, methods and practice and the relationships between these in information systems development?	The overall description of the contribution: Information systems development contexts influence and are influenced by competence, methods and practice and the relationships between these in information systems development. Contributions to answering research question 1 are presented in the following sections: 1. The "where and when" of deploying method 2. Establishing and maintaining a development team 3. Communication competence is crucial and challenging 4. Common guiding vision? 5. Realistic learning environments and learning processes 6. The importance of meta competence 7. Development-in-action

Table 10. Overview of research questions and contributions of this thesis

The literature review in chapter 2 found some descriptions and explanations of the relationships between competence, methods and practice in information systems development. The descriptions and explanations were found to be insufficient. This thesis therefore contributes to the research in information systems development by describing the relationships between competence, methods and practice from different perspectives that contribute to an understanding of the relationships.

5.1 Research Question 1

Q1. How do competence, methods and practice relate to and influence each other in information systems development?

The answer to this question is found in the totality of all five publications. Each publication provides part of the answer. The following statement is a general description of the answer.

This thesis provides: A description of influences and relationships between competence, methods and practice in information systems development.

The description is in itself a contribution to the information systems development field. As described in section 2.4.2, summarised in Table 5 and in Publication 4, the descriptions in the existing literature were found to be insufficient. This thesis therefore contributes to a better understanding of information systems development through relating the findings in chapter 4 to existing literature and showing how the findings contribute to understanding information systems development as described in sections 5.1.1 to 5.1.6.

5.1.1 Adding competence to methods and practice in information systems development

The literature review (chapter 2) found that relationships between methods and practice are well described (cf. Table 5). This thesis found that competence relates to both methods and practice. The focus of the course integration case (Publications 1 and 2) was on learning in order to increase students' competence. The findings presented in Publication 1 were based on a study environment where the students enacted their competence by deploying some methods in an enforced realistic practice. The students were enthusiastic about the study environment which provided realistic practice outside of university-invented projects. By analysing their enthusiasm, the researcher found, in retrospect, that enacting competence was an important part of the students' experience provided in the study environment. The researcher therefore concluded that there is a connection between competence, methods and practice.

The connection between competence, methods and practice was found in the government agency case (Publication 3), mostly based on incidents that did not work out as anticipated by the developers. The decoupling between "how to" deploy prototyping and "when and where to" deploy prototyping shows what happens when relationships between competence, methods and practice were not considered in information systems development (cf. section 4.3.2 the first finding). A lack of domain competence on the developers' side led to a learning process for the developers and reworking of the system before they succeeded in completing an information system for the government agency (cf. section 4.3.1 and section 4.3.2 the first finding).

The literature review (section 2.4.2) shows a tendency in the literature to focus too greatly on method alone, without fully considering competence and its role in information systems development. Taking the stand that competence, methods and practice form relationships in information systems development will emphasise the importance of letting all three elements inform information systems development. Thus adding competence to methods and practice contributes to a better understanding of what happens in information systems development.

Competence is mentioned as one element in method in action (Fitzgerald et al., 2002). However, Fitzgerald et al. (2002) do not discuss the relationships between competence and methods in detail. This thesis suggests that formally introducing competence into method in action will provide a vessel for explaining why the formal methods and the method in action differ as each developer, a unique person with his own unique competence, will probably deploy the same methods in practice differently from other developers.

Introducing competence in the emerging method (Madsen et al., 2006) may explain how and why the method emerges. As methods cannot deploy themselves, the deployer's view and understanding of the method will influence how the methods are deployed in practice. This thesis therefore suggests that one element in explaining the emerging methods is that the methods emerge through the developer enacting his competence when deploying the method in practice.

5.1.2 Influences exist between competence, methods and practice

In section 5.1.1, it was argued that competence be added to methods and practice in information systems development. The contribution described in section 5.1.2 gives a deeper understanding of how competence, methods and practice relate to each other. Changes in one of the elements of competence, methods and practice were found to influence the other elements, following three patterns: uni-directional influences, bi-directional influences and indirect influences. Uni-directional influence goes from one element directly to one of the two other elements. In the municipality case (Publication 4) the competence enacted by the development company negotiators in the negotiation process directly influenced their deployment of the method for winning the contract for developing a new information system.

Bi-directional influences were found to go both ways between two of the three elements. First the method influenced practice and then practice influenced method in the development process. In the municipality case (Publication 4) a developer experienced an unexpected result from his method deployment in practice. The developer said he would deploy the method differently next time due to the unexpected result.

The bi-directional influence may seem like an extension of the first type by being a uni-directional influence from one element to another element followed by another uni-directional influence between the same two elements but in the opposite direction. However, further analysis of the bi-directional influence suggests that the bi-directional influence seems to be a shortcut version of the third type of influence, the indirect influence. The indirect influence means that one element influences another element via the third element. Analysing the example of bi-directional influences described (above) suggests that the developer will not be able to deploy a method differently in the future if his competence is not changed. Due to the unexpected results of the developer's method when deployed in practice the developer had to reconsider his enacting of competence when deploying the method. The finding of the indirect influence explains why no uni-directional influence from method to competence was found in the reviewed literature (cf. Table 5 in section 2.4.2). This thesis suggests that the influence from method to competence went via practice to competence.

This thesis suggests that competence is important for the reflective developer's (Mathiassen, 1998; Mathiassen and Purao, 2002) ability to reflect and learn. However, competence is barely mentioned by Mathiassen (1998) and Mathiassen and Purao (2002). This thesis therefore suggests that competence and especially meta competence (Le Deist and Winterton, 2005) can explain why some reflective practitioners learn and change more than others, as competence and meta competence are important parts of a person's ability to reflect and learn.

Fitzgerald et al. (2002) discuss method in action and developer's competence. However, method in action is still the centre of the attention and research by Fitzgerald et al. (2002). This thesis suggests that competence, methods and practice influence each other more than Fitzgerald et al. (2002) suggest. The influences between the three elements in information systems development may explain why methods are deployed in action in different ways,

because the deployment is influenced by the developer's competence. Strengthening the developers' consciousness about competence enactment will influence method deployment, and the developers may have a better background for reflecting on and learning both from their own method in action deployment and the results of their deployment of method in action.

In the emerging method (Madsen et al., 2006), the role of the developer is recognised. The focus, however, is on how a local and unique method emerges in practice. Even if Madsen et al. (2006) recognise the importance of the developer in the emergent method, the enactment of developers' competence is not used as parts of the explanation of how and why the method emerges as it does in practice. This thesis therefore suggests that including an understanding of the influences between competence, methods and practice will add to explaining how and why the method emerges as it does as opposed to just concentrating on the emergence of method deployment in practice, as seems to be the focus of Madsen et al.'s (2006) explanation. Including competence as a third element together with methods and practice allows for explaining why and how competence emerges in information systems development. This thesis therefore suggests that the term emerging development describes better what actually happens in information systems development than the term emerging method (Madsen et al., 2006).

Influences between competence, methods and practice gives explanatory power to the four different versions of method (Ørvik et al., 1999). Since the developer has a central role in deploying methods the element of competence is very important in explaining how the developer understands a method and how the organisation deploys a method in information systems development. The developer's competence will therefore be important in both how he interprets and understand the method and how he deploys it in an actual information systems development.

5.1.3 Competence, methods and practice influence each other dynamically through their relationships

Section 5.1.2 explains that influences exist between competence, methods and practice through the relationships. This section provides three examples as a starting point for arguing that the relationships are dynamic. The dynamic influences between competence, methods and practice resulted in a learning process that increased the competence for one of the developers when he experienced an unexpected result in his deployment of a method in practice (cf. the municipality case). The dynamic nature of the influences was further seen in the decision to establish a development team to co-develop a module in the municipality's information system, basing the decision on earlier relationships between a developer in the development company and a sub-project manager in the municipality. The relationship between the two was deemed more important than their domain competence in the domain of the module to be developed. Through co-developing the module the developer and the sub-project manager from the municipality increased their domain competence. The dynamic nature of the influences was also observed when developers in the government agency case decided to redevelop the system after they had increased their domain competence through discussions and cooperation with the government agency.

The finding of dynamic changes in competence, methods and practice supports Madsen et al.'s (2006) critique of Fitzgerald et al.'s (2002) method in action for being too static when describing what happens in information systems development. However, the findings at the same time challenge, extend and widen Madsen et al.'s (2006) emergent method by

suggesting that all three elements, competence, methods and practice, are dynamically influencing each other through their relationships during information systems development. Madsen et al.'s (2006) emergent method is therefore too limited to explain what happens in information systems development, as it is not only the method that emerges. This thesis suggests that both competence and practice evolve dynamically through the relationships between them in information systems development. The main contribution from this thesis to Madsen et al. (2006) emerging method is that the developer and his competence must be considered in order to understand and possibly influence how the method emerges in information systems development. By focusing too much on the emergent method, Madsen et al. (2006) may overlook the importance of the developer's competence in information systems development.

Adding competence and practice to methods in information systems development changes the focus from the emergent method to emergent development, and contributes to method in action (Fitzgerald et al., 2002) by focusing on and including competence more explicitly than done by Fitzgerald et al. (2002). Even though method in action includes the developer, the developer's characteristics and the importance of the individual developer, the focus in the method in action model is still on the method. As the developer is the deployer of methods and he deploys method from his own competence and perspective, the method deployed in action will therefore be influenced by the developers' competence. The competence of the individual developer must therefore be considered when method is deployed to ensure a method deployment that furthers the information systems development instead of hindering it.

A consequence of the dynamic relationships between competence, methods and practice presented above was that consequently the information systems development emerged. It was not just the method that emerged (Madsen et al., 2006). However, the emergence of information systems development did not happen automatically. Findings in this thesis show that when the developers reflected on their experiences they were able to change their competence and their deployment of method in practice. Developers reflect and learn (Mathiassen, 1998) based on their experiences of enacting competence and deployment methods in practice. In the emerging information systems development suggested in this thesis it is assumed that competence, methods and practice will also emerge. However, when the developer does not reflect and learn from information systems development situations he may perform the same mistakes again and again. Mathiassen (1998) describes the reflective practitioner with a focus on reflection. Findings in this thesis suggest that in the reflection process all three elements, competence, methods, practice, their relationships and the dynamic nature of these needs to be considered and reflected upon. The reflection process further requires some competence in reflection to be able to reflect. The competence of reflecting is not discussed and proposed in Mathiassen's (1998) reflective practitioner or by Mathiassen and Purao (2002). This thesis therefore contributes to the work of Mathiassen (1998) and Mathiassen and Purao (2002) in that competence is an important element that is needed in order to reflect on what happens dynamically in information systems development and to learn from the reflections to be able to change both competence, methods and practice.

The dynamic influences between competence, methods and practice through their relationships shed light on Ørvik et al.'s (1999) different versions of a method (cf. section 2.2). Only the first version of the method is as the method maker/constructor intended it to be. The other three versions of method depend on how the method is understood and deployed. The deployment of any of the three versions of the method will therefore take place dynamically, in relation to and being influenced by which method is deployed and how the

developers enact their competence when deploying the method. Ørvik et al. (1999) state that it is the developer that interprets and deploys their version of the method. The dynamic relationships between competence, methods and practice presented in this thesis widen the understanding of the developer's interpretation and deployment of a method. The different versions of a method must therefore be understood and explained by what competence the developer has and how he enacts it. By considering the developer's competence it may also be possible to more deliberately plan how the developer's version of the method might be successfully deployed.

5.1.4 Competence, methods and practice form intrinsic relationships

Competence, methods and practice are found to exist in intrinsic relationships (Publication 4, cf. section 4.4.2 and sections 5.1–3). Information systems development is not possible if one of the three elements is removed in actual information systems development. The intrinsic relationships between competence, methods and practice are clearly seen in the indirect type of influences between the three elements in information systems development (cf. section 5.1.2), where influences from one of the elements go to any of the two others elements via the third element. The indirect type of influence is described in section 5.1.2. The description and arguments are therefore not repeated here. Instead examples of the existence of the intrinsic relationships in each of the three cases are presented in the following. The intrinsic relationships between competence, methods and practice presented in this section are clarifications and expansions of the contributions stated in sections 5.1.1–5.1.3. Competence is not just added to methods and practice in information systems development. Competence is an essential part of information systems development, together with methods and practice.

In the course integration case the students' study environment were set up with a forced realism where that the students had to enact their competence deploying methods when developing an information system for a company. All three elements, competence, methods and practice, were present in the study environment. The students' feedback pointed towards the enforced realism as an important part of the study environment.

The developers' lack of domain competence in the government agency domain led to reworking of the system, since the developers and the government agency's employees did not understand each other when they developed the requirement specifications for the new system. The lack of domain competence and not considering their lack of domain competence shows that the developers were not aware of the intrinsic relationship between competence, methods and practice. The developers' lack of awareness of the intrinsic relationships became visible when they reflected on their unsuccessful information systems development.

In the municipality's case, the decoupling of the "how to" competence and the "where and when" competence in prototyping led to unnecessary reworking since the developers were not able to deploy prototyping at an appropriate time to obtain the information they needed. The intrinsic relationships between competence, methods and practice were not recognised before the developers encountered problems in the information systems development.

The description of the intrinsic property of the relationships between competence, methods and practice is a contribution to existing literature. Many researchers (cf. Table 5, section 2.4.2) describe some of the relationships between competence, methods and practice. However, a description stating that the relationships are intrinsic was not found in information systems development research literature. The contribution from findings in this thesis to the information systems development theories of method in action (Fitzgerald et al., 2002), the

emergent method (Madsen et al., 2006), the reflective practitioner (Mathiassen, 1998; Mathiassen and Purao, 2002) and the four versions of a method (Ørvik et al., 1999) presented in section 5.1.3 are therefore not repeated here.

Instead the contribution in this section is related to other information systems development theory. Peppard et al. (2000) warn against mixing "the ability to…" in competence to the actual performance of competence. This thesis challenges the division of competence into the ability to and its actual carrying out. As competence, methods and practice form intrinsic relationships the three elements are closely connected in information systems development. It is therefore meaningless to state that competence is something different from the actual enactment of competence in practice. The intrinsic relationships between competence, methods and practice suggested in this thesis are in line with both Sandberg (2000), stating that the developer and development forms one entity in the actual carrying out of the development, and with Sandberg and Dall'Alba's (2009) entwinement of life with the world. It is not possible to split life, developer and development in actual information systems development. The intrinsic relationships further challenges Bassellier et al.'s (2003) duality of dividing what people do and what they possess. The intrinsic relationships between competence, methods and practice suggest that such duality does not describe what actually happens in information systems development.

Improved involvement and better personnel is more important than methods in information systems development (Necco et al., 1987). Necco et al.'s (1987) general statement about personnel echoes the suggestions in this thesis of an intrinsic relationship between competence, methods and practice. This thesis therefore clarifies some of Necco et al.'s (1987) meaning of better personnel, suggesting that developers and their competence are pinpointed as being important to produce better information systems.

Westrup (1996) states that developers construct their own understanding of the organisation and its context in information systems development. This thesis contributes to Westrup (1996) by stating that competence enacted by developers directly influences how they understand the organisation and thereby how they deploy methods in practice when developing information systems. Understanding the intrinsic relationships between competence, methods and practice in information systems development are therefore essential to be able to construct a realistic understanding of the organisation, the context and the information systems to be developed.

Information systems development may be influenced by shifts in the paradigm of the method or by deep-rooted paradigmatic values of the developers in the development (Päivärinta et al., 2010). This thesis contributes to Päivärinta et al. (2010) by suggesting that competence may help developers reflect on the influences that deep-rooted paradigmatic values may have on the information systems development. Recognising that competence, methods and practice form intrinsic relationships in information systems development may also open the developers mind to a wider consideration of the possible drift that may take place since the developers may notice changes in method deployment or unexpected results in practice of their enactment of competence and deployment of methods.

This thesis challenges Päivärinta et al.'s (2010) discussion of mismatches and drifts in method deployment. Mismatches and drifts in method deployment seem to assume that there may be one correct way of deploying methods. This thesis suggests that only machines are able to follow a method completely. As developers are not machines methods will always be deployed differently by different developers. The differences may be partly explained by the

intrinsic relationships between competence, methods and practice in information systems development as presented in this thesis in that the three elements dynamically influence each other through the relationships that exist in information systems development.

5.1.5 Competence and method are intertwined

Competence and methods were found to be intertwined (Publication 5), as described in two situations. A developer successfully deployed his competence in and experiences of parametric customisation to construct some standard parts of the information system that fit the municipality's size and needs. However it was difficult in the information systems development to state what was competence and what was method in the parametric customisation. Developers, together with the municipality, enacted their competence by deploying the development company standard systems development method when developing the invoicing module. Misunderstandings arose between the developers and the actors in the municipality. Challenging discussions took place and it was not possible to distinguish what was competence enactment or method deployment in the actual information systems development. The municipality then called a high-level meeting between the municipality and the development company to clarify the situation, sort out the problems and develop a plan to successfully solve the problem.

The two situations described, the parametric customisation and the development of the invoicing module, were similar in all aspects except two: the method deployed and the consequences of the method deployment for the municipality's actors. The differences in method and competence between the developer company and the municipality were the same in both situations. Parametric customisation was methodically handled successfully by the development company with no negative experiences for the municipality. When the developers deployed the development company's standard development method for developing the invoicing module the municipality experienced many negative consequences, leading them to push for the high-level meeting described above. When implementing the system the development company produced many test versions of the system that included errors that the municipality were unable to understand or handle.

In both situations the competence and the method described were intertwined to a degree that it was difficult for the researcher to distinguish what was competence and what was method. The developers in the development company were well accustomed in parametric customisation as they had to customise every standard information system they delivered to any municipality. As for the invoice module the municipality entered into a closer relationship with the development company's competence enactment and method deployment, as the municipality and the development company co-developed the module. The municipality was exposed, in a direct way, to the nitty-gritty parts of the development and mistakes made by the developers. It seems that the developers did not quite understand the effects of their competence enactment and method deployment on the municipality, and were therefore unable to handle the situations that arose.

The intertwinedness of competence and methods in information systems development deepens the understanding of the intrinsic relationships between competence, methods and practice. It is obvious from the examples presented and the contributions argued in sections 5.1.1–5.1.4 that a lack of consideration of one of the three elements in information systems development has consequences for the information systems development and for the results of the development. The contribution presented in this section is in line with Sandberg's (2000)

finding that worker and work form one entity, and his finding is therefore also valid in the information systems development field but was not introduced in the field earlier.

The intertwinedness between competence and methods seems to be ignored in research on deploying methods in information systems development. When Mathiassen and Purao (2002) discuss learning processes for reflective developers, the intertwinedness between competence and method is not discussed. This is strange because a reflective practitioner needs to reflect not only on the deployment of methods, but also on enactment of competence. Maybe the reason for the lack of describing the intertwinedness of competence and methods stem from Mathiassen and Purao (2002) only emphasising and relating the learning perspective to method. A result of the intertwinedness of competence and method suggested in this thesis is that the reflective developer (Mathiassen, 1998) needs to be aware of the intertwinedness when reflecting to make sure that he arrives at conclusions that can advise his learning process in a better way than just reflection on the method deployment alone.

The intertwinedness between competence and methods expands the "ability to …" (Bassellier et al., 2001) understanding of competence. Bassellier et al. (2001) warn of mixing the understanding of competence as "ability to …" and the actual enactment of competence. However, the intertwinedness between competence and method makes it difficult to distinguish between the ability to enact competence and the actual enactment of it, since the results of competence enactment are only seen in practice when methods are deployed in actual information systems development situations. This thesis therefore suggests that competence must be seen as something more than just being "able to" (Bassellier et al., 2001). The expanded understanding of competence may make it easier to improve method deployment since it also includes the importance of understanding that method deployment is very closely linked to competence enactment.

In every information systems development it is important to understand what is to be developed, the development situation and the context of the development. Since the understanding is dependent to a large degree on the individual developer (Cockburn, 2001), the individual developer will probably perceive and understand what is to be developed, the development situation and the context of the development differently from other developers. This difference in understanding will probably lead to different ways of deploying methods, as competence and method are intertwined in information systems development. This thesis therefore contributes to understanding Ørvik et al.'s (1999) four versions of a method, and why different developers will understand the method differently. Considering the different developers and their competence may give rise to understanding why the developers deploy method as they do. By understanding the developers' different deployment of methods it may be possible to interact with the developers to understand and influence their understanding of the method to be deployed and thereby the actual deployment.

Missing, or not giving attention to, the intertwinedness between competence and method may lead to a decoupling of competence and method in information systems development. One possible result of the decoupling may be that deploying method creates more problems in information systems development than it solves (cf. Publication 5).

5.1.6 Relationships exist between developer and information systems development

The contributions to information systems development research presented in sections 5.1.1 to 5.1.5 suggest that there is a relationship between developer and information systems

development. As shown in section 5.1.5 competence and methods are intertwined. Each individual developer enacts his unique competence when deploying methods in information systems development.

This thesis therefore suggests that there are relationships between developer and information systems development. In the information systems development literature reviewed in chapter 2 a description of a relationship between developer and development is lacking. Often methods and their deployment in practice are discussed and researched without including the developer as a person. The developer is often considered as having competence as an attribute for solving some given problems or producing some specific parts of an information system. One may ask if much of the existing literature on methods in information systems development want to eradicate the developer as a person by trying to explicate his unique competence and turn it into methods? Such perspective is challenging since its prerequisite will be that information systems development can be automatised and performed by computers without the interference of human competence. This thesis finds that information systems development includes creative ways of solving problems and intricate communication between actors.

This thesis posits that an understanding of the relationships between competence, methods and practice as presented in sections 5.1–5 presupposes that the developer is participating in the information systems development as a person, including his personality, his enactment of competence and his work capacity for deploying methods in practice. This thesis therefore suggests that the developer and information systems development are intertwined to such a degree that it is not possible to clearly distinguish what is the developer and what is information systems development work performed by the individual developer.

The relationships between developer and information systems development must be considered intrinsic. In other words it is meaningless to talk about information systems development without including the developer as a person, not only as an actor possessing competence, methods and experience of practice. Such understanding of the relationship between developer and information systems development is in line with the understanding of worker and work forming "one entity" in the experience of life (Sandberg, 2000, cf. section 2.4.1). Sandberg (2000) studies organisational behaviour and the education of workers. This thesis argues that the same understanding is valid in the information systems development field. This thesis therefore contributes to research in information systems development by suggesting that the relationship between developer and information systems development is intrinsic, meaning that none of them can be removed without harming information systems development.

The intrinsic relationship between developer and information systems development suggested in this thesis acknowledges that human beings in the world (Sandberg and Dall'Alba, 2009) offer themselves as persons, together with their competence and method deployment, into information systems development. The developers as persons cannot be separated from their work performance. This understanding of the relationship between developer and information systems development strengthens and broadens the criticism of the atomistic and fragmented understanding of competence and methods (Hager and Gonczi 1996) presented in section 2.1. The intrinsic relationship between developer and information systems development and the description of it presented in this thesis is therefore a contribution to a deeper and more balanced understanding of competence, methods and practice and their relationships in information systems development. Regarding the developer and information systems

development as forming an intrinsic relationship in information systems development shows that a lack of a specific competence in any given information systems development is not easily rectified by focusing on adding the competence seemingly lacking by employing some arbitrary developer with the desired competence. The totality of the information systems development, the developers already participating in the information systems development and the developers to be engaged to supply the competence needed have to be considered.

5.2 Research Question 2

Q2. How does context influence developers' competence, methods and practice, and the relationships between these, in information systems development?

The answer to this question is found in the totality of all five publications. Each publication provides part of the answer. The answer is described through one general statement that will be detailed through the contributions described in subsequent sections.

Information systems contexts influence and are influenced by competence, methods and practice and the relationships between these in information systems development.

Information systems are always developed in some context. The important tasks for the developers are to discover, understand, and handle the context of the information systems to be developed. Competence, methods and practice are important and central parts of information systems development. Their actual enactment and deployment in practice will influence and be influenced by the information systems development context.

5.2.1 The "where and when" of deploying method

This thesis suggests that competence in prototyping may be viewed at two levels, the "how to" level, i.e. deploying prototyping as a method, and the "when and where" level, i.e. when and/or where to deploy the method of prototyping. The relationships between the "how to" level and the "when and where" level seem to be decoupled in both the government agency case and the municipality case. The developers in both cases had competence in prototyping but reported that they deployed prototyping too late in the information systems development. They lacked the "when and where" level of competence of deploying prototyping. The "where and when" level competence requires that the developers understand and adapt closely to the context of the development, since "where and when" of prototyping depends to a large degree on the users of the system to be developed and their interaction with the developers, while "how to" competence, in contrast, is more dependent on the developer's knowledge of carrying out the technicalities of prototyping.

The contribution to the information systems development field presented in this section is that the developer has to consider the context of the development when deploying prototyping in information systems development. The contribution expands Ørvik et al.'s (1999) description of the four versions of a method. One of the four versions of the method is how the organisation deploys the method. This thesis suggests that the organisational context will influence how the method is deployed by the organisation. The finding therefore contributes to Ørvik et al. (1999) by explaining that organisational version of the method is influenced by the environments of the systems development. The influences from the environment must be considered and acted upon for understanding and deploying the organisational version of a method in information systems development. The contribution to the information systems development field presented in this section is in line with professional competence as ways of

being (Sandberg and Pinnington, 2009, cf. also Figure 3), that includes both the developer's self-understanding, understanding of the work of other people and of the tools related to the information systems development.

5.2.2 Establishing and maintaining a development team

Longstanding and good interpersonal relationships between developers from the development company and users in the municipality case were regarded as more important than the team's competence and methods experience when establishing a development team. The members of the development team increased their competence, while together they successfully developed modules for the new system.

The finding suggests that it is important to pay attention to contextual elements when development teams are organised and staffed. The team members joined the team bringing their professional competencies, their personal abilities, their former experiences and their "life-world perspective" (Sandberg and Dall'Alba, 2009). Some of these elements relate directly to the needs of the information systems development. Other elements come along with the persons selected to join the team. Based on the findings in this thesis, introducing a "life-world perspective" (Sandberg and Dall'Alba, 2009) into information systems development means that information systems development must include the context of the development as well as the development itself. The life-world perspective (Sandberg and Dall'Alba, 2009) adds an understanding of how the developers in a team may influence each other, the context of the development and the development itself. It is not enough to only consider what competence is to be present in the development team and how it is to be enacted because team members participate in team activities and they are and will therefore influence the other team members and their actions. The totality of the team will therefore constitute the team's competence in information systems development. This thesis therefore suggests that a team's performance will be influenced by how the team members act in the world, how others act in the same world and how things and artefacts are used in human ways of being (Sandberg and Pinnington, 2009) in any given information systems development.

In the government agency case the way development teams were established led to problems. Different contextual perspectives influenced the staffing of the teams. The government agency and the developer company agreed on dividing work to save money. The government agency should have developed the use cases that were to be the basis for developing the new information system (cf. section 4.3.1). Dividing work as described was not a good idea. The use cases were found to be too unspecific for developing the system. To rectify the problem a new development team was established consisting of team members from both the agency and the development company (cf. section 4.3.1). It is obvious that the government agency had not considered the information systems development context when they established their team.

Information systems development teams may be established from a learning perspective. In the course integration case, suggestions on how to organise and integrate courses (Publication 2) were made from a learning perspective, intentionally designed so that the student groups had to experience the results of their own work in the information systems development learning process. Designing the study environments in that way allows for reflection (Mathiassen 1998) in the students' learning processes, both individually and in teams. Through the reflection the students get opportunities to increase their competence. A contribution from the study environment case emphasises (Publication 1) that realism felt by the student groups which increased their motivation for learning. The increase in motivation

then became a prerequisite for the students to engage in the project provided through the study environment. As the students had to work in teams the team fit can contribute to the understanding of the information systems development context and provide motivation for the learning effort that the students needed to exercise.

The information systems development team is the main contact point between the information system in developing and the context where the information system is to be used. It is therefore important to establish and dynamically maintain a development team that can serve the information systems development from start to finish. This concerns both how the relationships between the development team and the development context are dynamically developed, and how the development team is itself organised throughout the development. The finding presented in this section provides empirical data to Green's (1989) study of how developers relate to end users. The finding contributes to a wider understanding of competence, as Green (1989) atomised competence. This thesis suggests that competence has to be viewed together with method, practice and the context of the information systems development to understand and maintain the relationship between developers and users.

5.2.3 Communication competence is crucial and challenging

Findings in this thesis suggest that general communication competence does not suffice in information systems development. In the government agency case the developers had general communication competence, but yet were not able to understand the domain of the government agency. It took some time before the developers understood that they did not understand the government agency and its domain and the consequences the domain had for the information systems development. In the municipality case a developer from the development company and an actor from the municipality sat together in front of a PC, discussed and tested prototypes as they worked together to develop an e-commerce solution for the municipality. Through this cooperation they developed an understanding of the domain and the domain's specific language.

The two situations described suggest that the communication needs to be specific and adjusted to the domain where the system is to be used and to the context of the information systems development. The developers and the users in both cases had general communication competence. However the developers' communication competence was not sufficient to discover the misunderstandings that arose (cf. sections 4.3.2 and 4.4.2). Communication is critical in any information systems development process (Cockburn, 2001; Fitzgerald et al., 2002; Mathiassen and Purao, 2002). However earlier research on communication in information systems development often only states that communication is critical for information systems development success without going deeper into the role domain specific communication competence may play. An exception is Cockburn (2001) that gives much attention to communication in information systems development. However Cockburn (2001) is mainly occupied with the technicalities of communication.

This thesis suggests that communication competence in information systems development needs special attention as it is the most important vessel connecting the context of the development to the actual information systems development. Through the vessel of communication the developer brings his whole being into the information systems development situation. This thesis therefore agrees that the developers' understanding of work, of the context, of others and of the methods to deploy (Sandberg and Pennington, 2009) will constitute their distinct form of communication competence when developing information systems. Understanding the consequences of developers' being in the world as

persons (Sandberg and Dall'Alba, 2009) can therefore be applied in the information systems development field.

Communication competence is a basic competence for learning in information systems development. In all three cases presented in this thesis the developers were dependent on information from the users to be able to understand the domain and the information systems' requirements. Through interaction with the users the developers learnt about the domain and were able to carry out their development activities in a meaningful way. A common learning process took place where both the users and the developers learnt more about the domain and the context for the development. Their learning took place through reflecting (Mathiassen, 1998; Mathiassen and Purao, 2002) on the situations that arouse. The findings in this thesis add to Mathiassen (1998) and Mathiassen and Purao (2002) thoughts on the importance of domain specific communication competence. If the reflective practitioner (Mathiassen, 1998) does not understand the domain he is to develop an information system for he may likely fail in his development efforts. This thesis therefore suggests that it is necessary to add the importance of domain competence to the competences to be enacted by the reflective practitioner (Mathiassen, 1998). The reflective practitioner needs to broaden his understanding of the communication challenges and difficulties in information systems development to obtain a wider basis for his reflections through understanding the importance of communication in information systems development.

Domain competence in all three cases in this thesis relate directly to the learning found to take place in each of the three cases. The learning that took place depended on the experiences related to the information systems development itself and experiences related to the context of information systems development. It seems important to distinguish between experiences related to the inside of the project exemplified with the "how" of prototyping (cf. findings both in the government agency case and the municipality case) and experiences related to the contextual environment of the project, exemplified by the "when and where" of prototyping, to understand the arena for learning in information systems development. The "how to" of prototyping relates to enacting competence deploying method to make a prototype. Learning "how to" is closely related to the technicalities of the method of prototyping. Learning in the "when and where" of prototyping in contrast requires a greater openness to understanding the context of the development and the needs of the context and the users of the system. Much of the literature in the information systems development field describes the "how to" of learning a method and the result of method deployment. However no other research is found in information systems development that describes the influence the context has on learning through the relationships between the competence, methods and practice related to the where and when of deploying methods in information systems development. The importance of domain competence for learning about the context in information systems development is therefore a general contribution to the information systems development field.

5.2.4 Common guiding vision

This thesis challenges Madsen et al.'s (2006) view that a "common vision" is better in guiding information systems development projects than strict adherence to a method (Madsen et al., 2006). Analysis of data in the three cases leads to a common conclusion that it is not easy to communicate a vision clearly enough to allow everybody understand it as a common vision.

In the course integration case the students found that it was difficult to understand the needs of the companies for whom they were to develop information systems. In the government agency the developers worked for some time before they realised that they had misunderstood

the government agency actors, and in the municipality case there were many misunderstandings, some of which were only clarified when they were taken to a higher organisational level in the development organisation. Findings in the government agency show how three different documents made the systems development difficult, since the content and conditions in each of the documents were not consistent.

In the examples given it took time and effort through communication to get to a common understanding of the situations at hand. It is therefore reasonable to anticipate that simply making a common vision that is understood by all the developers and the other actors in an information systems development is a huge and complex task. The complexity concerns the technicalities of communication, the vision itself, getting the developers and other actors to accept what is communicated and understood of the common vision, and to work according to the common vision in the actual information systems development. The developers and the other participants in the information systems development bring their different backgrounds, competence, methods, and experiences in practice and their own perspectives on the actual development situation and its context. They may also have their own agenda on what they want, both from the information systems development and from the finished system. The contribution suggested in this section is just to problematise a common vision (Madsen et al., 2006) as a guiding principle in information systems development. Suggestions for a solution to the challenge are outside the work of this thesis.

5.2.5 Realistic learning environments and learning processes

Learning processes are described and discussed in all five publications. Learning was studied in dynamic information systems development situations. In all three cases learning was found to take place in the dynamic, intrinsic and intertwined relationships between competence, methods and practice, and was also found to relate to the development context. In the course integration case the students obtained feedback on their work and learnt through the feedback they received and acted on. In the government agency case the developers started to learn the domain competence by experiencing problems at the start of the information systems development. When the developers understood that they did not understand the government agency actors' domain they learnt the domain competence through communication and by working together with the actors of the government agency to develop the system. In the municipality case a developer reported that he learnt through obtaining unexpected results from deploying a method.

In the three examples mentioned learning did not happen automatically. Learning is related to reflections done by the developer or practitioner (Mathiassen, 1998; Mathiassen and Purao, 2002) and may take place in situations where competence, methods and practice are in play. Learning processes are described in general in the educational literature in the information systems development field, but are not described specifically in relation to competence, methods and practice. Mathiassen and Purao (2002) describe the reflective practitioner and the experiential learning that takes place through deploying methods in practice. However Mathiassen and Purao (2002) do not describe the importance that the influences between the three elements of competence, methods and practice and between the three elements and the development context have for learning.

The contribution from this thesis to understanding learning in information systems development is that competence, methods and practice, the relationships between them and their relationships to the development context, need to be considered and reflected upon in the learning process; in organised forms in courses for students or in actual information systems

development performed by more or less experienced systems developers. By reflecting on the three elements, competence, methods and practice, the learner will have to consider influences in learning from the three elements and may therefore learn different things and in a different way than just focusing on the method and its deployment in practice. Bloom (1956) describes different levels of learning competence from the lowest level, knowledge, to the highest level, evaluation. Benner (1984) describes different levels for the developer to develop competence, starting from novice and ending at expert. This thesis suggests that their views of competence need to be introduced into information systems development. It seems that the information systems development literature is less concerned with the level of competence and just assumes that the developers have a certain level of competence in information systems development.

5.2.6 The importance of meta competence

This thesis suggests than one way of explaining the learning process that takes place in information systems development is through the term meta competence (Le Deist and Winterton, 2005). Le Deist and Winterton (2005) criticise the prevailing narrow understanding of competence and expand on the competence concept by an element they call "meta competence" (cf. Figure 1). Literature in information systems development generally describes three types of competence: competence, knowledge and skills (cf. section 2.1.2). As shown in Tables 3, 4 and 5 the competence concepts, knowledge concepts and the skills concepts are all of a practical type ranging from technical competence via process orientated competence to the how to do competence. The meta competence perspective is lacking in the presentation of competence in the information systems development field described above.

This thesis suggests that in information systems development meta competence means being able to and actually reflecting on what is happening in development practice, learning from the reflections, acquiring new competence and eventually changing the direction of the information systems development through changes in either the individual developer, in the team or in the way competence is enacted and methods deployed in practice. Meta competence is therefore an extension of the ordinary understanding of the competence concept in information systems development that enables the learning and enacting of other competencies in information systems development. The enactment of meta competence is therefore a contribution to understanding the dynamic characteristics of the emergent method (Madsen et al., 2006).

For the reflective practitioner (Mathiassen, 1998; Mathiassen and Purao, 2002) reflection is an important part of the practitioner's way of learning and continually improving his/her performance. When Mathiassen (1998) and Mathiassen and Purao (2002) discuss the reflective practitioner they indirectly suggest meta competence, but do not explicate it in their work. Adding meta competence to their model of the reflective practitioner means that the learning processes that take place through the reflection performed by the reflecting practitioner expand their understanding of learning, and of how reflection can help them to change their ways of working.

5.2.7 Development in action

Concluding the above discussion (sections 5.2.1–5.2.6) on the influences between the context and competence, methods and practice in information systems development, this thesis suggests the term "development in action" rather than "method in action" (Fitzgerald et al., 2002) to describe what actually happens in information systems development. The term development in action includes competence, methods and practice, the relationship between

them and the totality of the context for an information systems development. The arguments for the term development in action follow.

This thesis has already argued that the developer and the information systems development are intrinsically related to each other in information systems development and form one entity in the experience of information systems development (cf. section 5.1.6). The intrinsic relationship influences the whole information systems development in a dynamic and intertwined manner. The influence is wider than that described in the emergent method (Madsen et al., 2006) or method in action (Fitzgerald et al., 2002). Understanding the intrinsic relationship between developer and information systems development may contribute to understanding and explaining successes and/or failures in information systems development. As shown in this thesis successes and/or failures in information systems development depend on more than just deploying methods in practice. The developer's understanding of the task he is to solve and his own attitude both to the task and to enacting competence and deploying methods will influence the design and the outcome of the systems development process (cf. section 5.1).

The understanding of the one entity formed between developer and information systems development was inspired by Sandberg's (2000), finding of one entity between workers and work in an organisational theory context. Introducing the understanding of the "one entity" into information systems development is a way of relating research results in organisation theory with information systems development.

The one entity that forms between developer and development described above is on an individual level. Development in action, however, also includes the team perspective and the contextual perspective on information systems development. The team of developers and other actors will form one entity in information systems development. The entity formed by a team will be more dynamic that the entity formed by one developer, because the team includes more developers that actively participate in the entity formed and both the individual team members and the number of teams may change over time. The one entity formed by the team and the information systems development will be influenced and influence the four elements of professional competence (cf. Figure 3 section 2.4.1): self-understanding, the understanding of the work, of other people and of tools (Sandberg and Pinnington, 2009). Together these four elements lead to distinct forms of competence in work performance (Sandberg and Pinnington, 2009) both for the individual developer and for a team of developers and other actors. A close look at the four elements of professional competence reveals that some of the elements that lead to professional competence come from the context of information systems development, namely self-understanding, the understanding of other people and partly the understanding of the work in general. The four elements of professional competence are included in the concept of development in action.

The term development in action further answers the criticism raised towards the concept of competence, often described as being too atomistic (cf. section 2.4.2). The term development in action includes everything that happens in information systems development. Competence, methods and practice are related to all parts of information systems development, including the context of the development. It is therefore impossible to limit competence and method just to the more technical enactment of competence and deployment of methods in practice.

The term development in action therefore includes more of the totality of the information systems development situation than encompassed in methods in action (Fitzgerald et al.,

2002) or the emergent method (Madsen et al., 2006). In the two theories mentioned (Fitzgerald et al., 2002; Madsen et al., 2006) the emphasis is on methods with a connection to practice. Both theories give the developer some role in the development, but do not describe the relationship between the developers' competence and the information systems development. From the above discussion it is evident that competence enacted and methods deployed in practice and the influences between them are essential elements in any information systems development. However, the influences from the context of the information systems development are equally important and will influence both how the three elements of competence, methods and practice influence each other and how they in turn will influence the context. Development in action is therefore a term that better describes what actually takes place in information systems development. The content of the term development in action may also shed light on the different versions of methods deployment described by Ørvik et al. (1999), by including the wide competence concept included in development in action.

6 Conclusion

This thesis explores the development of information systems through two research questions. The first research question studied the three elements of competence, method and practice and the relationships between these in information systems development. The second research question was concerned with the study of how the three elements of competence, methods and practice and the relationships between them influenced and were influenced by the different practical contextual arrangements of information systems development.

The two research questions were first addressed by studying the literature presented in chapter 2. They were further studied through the three case studies presented in chapter 3. The research methods used were questionnaires, document studies and interviews. The three cases were used to collect data for the five publications presented in chapter 4. Answers to the two research question were found by using contributions from all five publications. The contributions are presented, discussed and related to existing theory in information systems development in chapter 5.

In this chapter, first the contributions from this thesis to knowledge in information systems development are described. The chapter then reviews the implication for practice, discusses the limitations of the contributions described in this thesis, and ends with suggestions for further research.

6.1 Contribution to knowledge

The first research question was: How do competence, methods and practice relate to and influence each other in information systems development? The main contribution from this research question to knowledge in information systems development is that a systems developer and systems development form an intrinsic relationship, and as such form one entity in information systems development (cf. section 5.1).

The main contribution to knowledge is based on the following findings and contributions. This thesis argues that competence must be introduced into the relationships between method and competence. The three elements competence, methods and practice were found to influence each other through the relationships between them in information systems development. The influences were found to follow three different patterns: uni-directional, bi-directional and indirect influences between the three elements (cf. section 5.1.2). The indirect influence was not found to have been described in research in information systems development literature. The influences were further found to be dynamic. A consequence of the dynamic nature of the influences between competence, methods and practice was that the three elements were found to form intrinsic relationships, meaning that all three need to be present in information systems development. Removing or not considering one of the elements of competence, methods or practice in information systems development will have a negative effect on the development. A consequence of the intrinsic relationships is that the developers cannot exceed the limitations that are established through their competence, methods and practice. This is obvious through the intertwined nature of the relationships found between competence and method.

The second research question was: How does context influence developers' competence, methods, and practice and the relationships between these in information systems development? This thesis finds that contextual arrangements in information systems

development both influence and are influenced by developers' competence, methods and practice and the relationships between them. The contextual arrangements were found to influence information systems development in different ways, from improving practice, hindering practice or being counterproductive.

The most important part of the contribution to research question 2 is the term development-in-action. The term is an expansion of the one entity that is formed between the developer and development in information systems development, as described within research question 1, to include other developers and context in information systems development. Information systems development takes place both inside the development team, in relation to the users of the information system to be developed and in relation to other contextual arrangements, like the company that will use the information system, its deployment of method and the agreements and contracts negotiated for the development of the system.

The contribution to knowledge from research question 2 includes a strong criticism of the prevailing understanding of competence found in the literature review (cf. chapter 2), where the concept of competence in information systems development was found to be too atomistic, too rationalistic and context-free. The contribution from this thesis is that context is an important factor in understanding and enacting competence in information systems development.

This thesis suggests that meta competence needs to be included in the concept of competence in information systems development. Meta competence includes reflection on the other competencies enacted in information systems development and the acquisition of new competencies.

6.2 Implications for practice

The findings in this thesis have implications on practice in three different areas: designing information systems development methods, designing information systems development processes and their contexts, and staffing information systems development teams.

The findings in this thesis suggest that method designers need to consider the relationships between competence, methods and practice when designing information systems development methods. An understanding of the intrinsic, dynamic and intertwined relationships between competence, methods and practice and how these elements influence each other in information systems development may aid method designers when considering what is required to deploy their methods. A method in itself has little value if it is not deployed in practice. The deployment of methods depends completely on the developers that enact their competence. It is therefore important for method designers to understand and consider what competencies need to be performed for the successful deployment of methods in practice when the designers design systems development methods.

The information systems developer and the development team must consider the competence, methods and practice present in information systems development, the relationships between the three components and the actual context for the development when the developers decide how to develop an information system. Competence is an important element in the construction of development practice, as in any case the individual developers enact their competence to contextualise the method.

Staffing information systems development teams will therefore be important. Therefore what competence is needed and what methods need to be considered when staffing information systems development teams? How will the information systems development team members, with their differences in competence and experience in methods deployment in practice, be able to enact their competence and deploy the methods that will advance the information systems development processes? This thesis suggests that the developer and the development form an intrinsic relationship, understood as 'one entity' (Sandberg, 2000), in information systems development. A developer does not only participate in an information systems development team as a method deployer or a competence enactor, but as a person with all his/hers personality, competence, methods competence and experience of deploying methods in practice.

6.3 Limitations

The validity and limitations of data collection and data analysis activities were discussed in chapter 3. The limitations presented here concern generalisability. Since the research approach in this thesis is explorative, describing the phenomenon sought in the research questions is more important than searching for the generalisability of the answers to the research questions. According to Creswell (2007, p. 76) generalisability holds little meaning for qualitative researchers. This thesis provides a description of the relationships between competence, methods and practice in information systems development. However, this thesis also provides suggestions for method and information systems development designers to consider. The suggestions are not generalisable but may still be valuable in the future in other information systems development projects and for their contexts (Walsham, 1995).

According to Strauss and Corbin (1998), theory-building in qualitative research is concerned with "talking more the language of explanatory power rather than of generalizability" (p. 267). "Explanatory power means 'predictive ability', that is the ability to explain what might happen in given situations" (Strauss and Corbin, p. 267). This thesis describes the relationships between competence, methods and practice and their influences. The description of the relationships contains explanations of phenomenon that have not previously been researched in the information systems development field. The description of the relationships between competence, methods and practice and the context of where information systems development takes place also includes elements explaining the relationships and how the elements influence each other. Research results in qualitative research are difficult to generalise, and care should be taken in such attempts. However, the results of the research described in this thesis may be valuable for use in future information systems development, and as enlarging the understanding and explanations of what happens in information systems development.

6.4 Further research

The answers to the two research questions posed in this thesis call for further research. Such research may take different paths. One theoretical path may be to analyse existing systems development methods descriptions of competence, methods and practice and their use in the method. Such a study may reveal how the method designers consider competence and practice when designing methods, and what advice the designers attribute to competence and the need for competence for deploying the methods.

An empirical path may be used to study the actual deployment of methods and compare them with the results of studying methods design theoretically, as described above. Such a

comparison may be made both through case studies of information development in process and by studying the descriptions of completed information systems developments.

An actions research path may be employed to study actual information systems development projects from the perspective of ‘one entity’ (Sandberg, 2000) between the developer and development. Such study will include the developers’ competence and attitude towards the domains for which they develop information systems. Through the use of an action research approach the researcher may influence the developers and the development over time by providing learning opportunities for the developers through their actual information systems development practice.

References

Anderson, L. W. and Krathwohl, D. R. e. (2001). A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives. New York, Longman.

Avison, D. E. and Fitzgerald, G. (1995). Information Systems Development: Methodologies, Techniques and Tools, McGraw-Hill Book Company, London.

Bassellier, G. and Reich, B. H. (2001). Information Technology Competence of Business Managers: A Definition and Research Model. Journal of Management Information Systems 17 (4): 159-182.

Bassellier, G. and Benbasat, I. (2004). Business competence of information technology professionals: conceptual development and influence on it-business partnerships. MIS Quarterly, 28 (4): 673-694.

Bassellier, G., Benbasat, I. and Reich, B.H. (2003). The Influence of Business Managers' IT Competence on Championing IT. Information Systems Research, 14 (4): 317-336.

Benbasat, I., Goldstein, D.K. and Mead, M. (1987). The Case Research Strategy in Studies of Information Systems. MIS Quarterly, 11 (3): 369-386.

Benner, P. (1984). From Novice to Expert. Excellence and Power in Clinical Nursing Practice. Addison-Wesley Publishing Company Inc.

Bjerknes, G. and Bratteteig, T. (1995). User Participation and Democracy: A Discussion of Scandinavian Research on System Development. Scandinavian Journal of Information Systems 7 (1): 73-98.

Bloom, B. S. e. (1956). Taxonomy of Educational Objectives, the classification of educational goals – Handbook I: Cognitive Domain. New York, McKay.

Boyatzis, R. E. (1982). The Competent manager: a model for effective performance. New York, John Wiley.

Braa, K. and Vidgen, R. (1999). Interpretation, intervention, and reduction in the organizational laboratory: A framework for in-context information systemsm research. Accounting, Management and Information Technologies 9 (1): 25-47.

Brinkkemper, S. (1996). Method engineering: engineering of information systems development methods and tools. Information and Software Technology 38 (4): 275-280.

Brooks, F. (1987). No silver bullet: essence and accidents of software engineering. IEEE Computer Magazine, (April): 10-19.

Brown, J.S. and Duguid, P. (1991). Organizational Learning and Communities-of-Practice: Toward a Unified View of Working, Learning, and Innovation. Organization Science 2 (1): 40-57

Ciborra, C. U. (1998). Crisis and Foundation: An inquiry into the Nature and Limits of Models and Methods in the Information Systems Discipline. Journal of Strategic Information Systems, (7): 5-16.

Ciborra, C. U. (2002). The Labyrinths of Information, Challenging the Wisdom of Systems. Oxford University Press, New York,.

Cockburn, A. (2001). Agile software development. Addison-Wesley, Boston, Mass.

Creswell, J.W. (2003). Research Design Qualitative, Quantitative, and Mixed Methods Approaches, Second Edition. Sage Publications.

Creswell, J.W. (2007). Qualitative Inquiry and Research Design. Choosing Among Five Approaches. Second Edition. Sage Publications.

Curtis, B., Krasner, H. and Iscoe, N. (1988). A field study of the software design process for large systems. Communications of the ACM 31 (11): 1268-1287.
Eisenhardt, K.M. (1989). Building Theories from Case Study Research. The Academy of Management Review, 14 (4): 532-550.
Feeny, D. F. and Willcocks, L. P. (1998a). Core IS Capabilities for Exploiting Information Technology. Sloan Management Review 39 (3): 9-21.
Feeny, D. F. and Willcocks, L. P. (1998b). Re-designing the IS function around core capabilities. Long Range Planning 31 (3): 354-367.
Fitzgerald, B. (1996). Formalized systems development methodologies: A critical perspective. Information Systems Journal 6 (1): 3-23.
Fitzgerald, B. (1997). The use of systems development methodologies in practice: a field study. Information Systems Journal 7 (3): 201-212.
Fitzgerald, B., Russo, N. L. and Stolterman, E. (2002). Information Systems Development: Methods in Action. McGraw-Hill Companies.
Fitzgerald, B., Russo, N.L. and O'Kane, T. (2003). Software Development Method Tailoring at Motorola. Communications of the ACM 46 (4): 65-70.
Fitzgerald, B., Hartnett, G. and Conboy, K. (2006). Customising agile methods to software practices at Intel Shannon. European Journal of Information Systems 15 (2): 200–213.
Garavan, T. N. and McGuire, D. (2001). Competencies and workplace learning: Some reflections on the rhetoric and the reality. Journal of Workplace Learning 13 (3-4): 144-164.
Glaser, G.I. and Strauss, A.L. (1967). The discovery of grounded theory: strategies for qualitative research. Aldine, Chicago
Green, G. I. (1989). Perceived Importance of Systems Analysts' Job Skills, Roles, and Non-Salary Incentives. MIS Quarterly June: 115-133.
Hager, P. and Gonczi, A. (1996). What is competence? Medical Teacher 18 (1): 15-19.
Harmsen, F., Brinkkemper S., and Oei, H. (1994). Situational Method Engineering for Information System Project Approaches. IFIP Transactions Vol. A-55: 169-194. Amsterdam, North-Holland,
Huisman, M. and Iivari, J. (2002). The individual deployment of systems development methodologies. CAiSE 2002: 134-150, Springer-Verlag Berlin.
Iivari, J. (1994). Object-oriented information systems analysis: Comparative analysis of six object oriented methods. IFIP Transactions Vol. A-55: 85-110.
Iivari, J. and Hirschheim, R. (1996). Analyzing information systems development: A comparison and analysis of eight development approaches. Information Systems Journal 21 (7): 551-575.
Iivari, J. and Hirschheim, R. (1998). A paradigmatic analysis contrasting information systems development approaches and methodologies. Information Systems Research 9 (2): 164-193.
Iivari, J., Hirschheim, R. and Klein, H.K. (2000). A Dynamic Framework for Classifying Information Systems Development Methodologies and Approaches. Journal of Management Information Systems 17 (3): 179-218.
Iivari, J. and Lyytinen, K. (1998). Research on Information Systems Development in Scandinavia – Unity in Plurality. Scandinavian Journal of Information Systems, 10 (1-2): 135-186.
Jayaratna, N. (1994). Understanding and evaluating methodologies. NIMSAD: A Systemic Framework. McGraw-Hill Book Company.
Kautz, K., Madsen S. and Nørbjerg, J. (2007). Persistent Problems and Practices in Information Systems Development. Information System Journal 17(3): 217-239.

Keil, M., Mann, J. and Rai, A. (2000). Why Software Projects Escalate: An Empirical Analysis and Test of Four Theoretical Models. MIS Quarterly 24 (4): 631-664.
Kiely, G. and Fitzgerald, B. (2003). An investigation of the use of methods within information systems development projects. IFIP WG 8.2, Athens Greece.
Le Deist, F. o. D. and Winterton J. (2005). What Is Competence? Human Resource Development International 8 (1): 27-46.
Lee, D. M. S., Trauth, E. M. and Farwell, D. (1995). Critical Skills and Knowledge Requirements of IS Professionals: A Joint Academic/Industry Investigation. MIS Quarterly 19 (3): 313 - 340.
Lindgren, R., Henfridsson, O. and Schultze, U. (2004). Design principles for competence management systems: A synthesis of an action research study. MIS Quarterly 28 (3): 435-472.
Lucas, H.C. (1975). Why Information Systems Fail. Columbia Univ. Press, New York, 1975
Madsen, S., Kautz, K. and Vidgen, R. (2006). A framework for understanding how a unique and local development method emerges in practice. European Journal of Information Systems 15: 225-238.
Mathiassen, L. (1998), Reflective systems development, Scandinavian Journal of Information Systems, 10 (1-2): 67-118.
Mathiassen, L. and S. Purao (2002). Educating reflective systems developers. Information Systems Journal 12 (2): 81-102.
Munro, C.M., Huff, S.L., Marcolin, B.L. and Compeau, D.R. (1997). Understanding and measuring user competence. Information & Management 33 (1): 45–57
Nielsen, P. A. (1989). Reflections on Development Methods for Information Systems A Set of Distinctions Between Methods. Office: Technology & People 5 (2): 81-104.
Nielsen, P. (1991). Learning and Using Methodologies in Information Systems Analysis and Design. Ph D thesis Lancaster University.
Necco, C. R., Gordon, C. L., and Tsai, N. W. (1987). Systems Analysis and Design: Current Practices. MIS Quarterly 11 (4): 461-478.
Nordhaug, O. (1998). Competence Specificities in Organizations. International Studies of Management & Organization 28 (1): 8-29.
Olle, T.W., Sol, H.G. and Tully, C.J. (Eds.) 1983. Information Systems Design Methodologies: A Feature Analysis. Amsterdam, Netherlands: Elsevier Sci. Publications.
Olle, T.W., Sol, H.G., Tully, C.J. and Verrijn-Stuart, A.A. (Eds.) 1986. Information systems Design Methodologies: A Comparative Review. Amsterdam, New-Holland
Patton, M.Q. (1990). Qualitative evaluation and research methods. Sage,Newbury Park, California.
Päivärinta, T., Sein, M.K. and Peltola, T. (2010). From ideals towards practice: paradigmatic mismatches and drifts in method deployment. Information Systems Journal 20: 481-516.
Peppard, J., Lambert, R. and Edwards, C. (2000). Whose job is it anyway?: Organizational information competencies for value creation. Information Systems Journal 10 (4): 291-322.
Prahalad, C. K. and Hamel G. (1990). The Core Competence of the Corporation. Harvard Business Review 68 (3): 79-91.
Robey, D. and Markus, M. (1984). Rituals in Information System Design. MIS Quarterly. 8 (1): 5-15.
Robey, D., and Newman, M. (1996). Sequential Patterns in Information Systems Development: An Application of a Social Process Model, ACM Transactions on Information Systems 14 (1): 30-63.

Russo, N. L. and Stolterman E. (2000). Exploring the assumptions underlying information systems methodologies. Information Technology and People 13 (4): 313-327.
Sandberg, J. (2000). Understanding human competence at work: An interpretative approach. Academy of Management Journal 43 (1): 9-25.
Sandberg, J. and Dall'Alba, G. (2009). Returning to Practice Anew: A Life-World Perspective. Organization Studies 30 (12): 1349-1368.
Sandberg, J. and Pinnington, A.H. (2009). Professional Competence as Ways of Being: An Existential Ontological Perspective. Journal of Management Studies 46 (7): 1138–1170.
Sauer, C. (1993). Why information systems fail: a case study approach. Alfred Waller Limited, Henley-on-Thames.
Siau, K. and Rossi, M. (1998). Evaluation of Information Modelling Methods - A Review. HICCS (5): 314-322.
Siau, K., and Tan, X. (2005). Evaluation Criteria for Information Systems Development Methodologies. Communications of the Association for Information Systems 16: 860-876.
Sommerville, I. and Kotonya, G. (1998). Requirements Engineering: Processes and Techniques. New York, John Wiley & Sons Inc.
Simpson, B. (2002). The Knowledge Needs of Innovating Organisations. Singapore Management Review 24 (3): 51-60.
Smolander, K., and Rossi, M. (2008). Conflicts, Compromises, and Political Decisions - Methodological Challenges of Enterprise - Wide E-Business Architecture Creation. Journal of Database Management, 19 (1): 19-40.
Strauss, A. and Corbin, J. (1998). Basics of Qualitative Research, Techniques and Procedures for Developing Grounded Theory, Second Edition. Sage Publications.
Taylor, F. W. (1911). The Principles of Scientific Management. New York, HarperCollins and Norton.
Truex, D., Baskerville, R. and Travis, J. (2000). Amethodical Systems Development: The Deferred Meaning of Systems Development Methods. Accounting, Management, and Information Technologies 10: 53-79.
Walsham, G. (2006). Doing interpretive research. European Journal of Information Systems, 15: 320-330.
Walz, D.B., Elam, J.J. and Curtis, B. (1993). Inside a Software Design Team: Knowledge Acquisition, Sharing, and Integration. Communication of the ACM 36 (10): 63-77.
Wastell, D. (1995). The fetish of technique: methodology as a social defense. Information Systems Journal 6 (1): 25-40.
Westrup, C. (1996). Transforming Organizations Through Systems Analysis: Deploying New Techniques for Organizational Analysis in IS Development. In: Information Technology and Changes in Organizational Work. Orlikowski, W. J., Walsham, G., Jones, M. R. and J. I. DeGross (eds.). Chapman and Hall, London, pp. 157-176.
Vidgen, R. (2002). Constructing a web information system development methodology. Information Systems Journal 12 (3): 247-261.
White, K. B. and Leifer R. (1986). Information systems development success. Perspectives from project team participants. MIS Quarterly 10 (3): 214-223.
Willcocks, L., Feeny, D. and Olson, N. (2006). Implementing Core Capabilities: Feeny-Willcocks IT Governance and Management Framework Revisited. European Management Journal 24 (1): 10.
Webster, J. and Watson, R.T. (2002). Analyzing the Past to Prepare for the Future: Writing a Literature Review. MIS Quarterly 26 (2): 13-24.
Yin, R. K. (2009). Case Study Research. Design and methods. 4. edition). Sage.

Zalkowitz, M., Yeh, R., Hamlet, R., Gannon, J. and Basili, V. (1984). Software engineering practices in the U.S. and Japan. IEEE Computer 17 (6): 57-66.
Ørvik, T. U., Olsen, D. H. and Sein, M. K. (1999). Deployment of System Development Methods. Exploring Paradigmatic Mismatches. Evolution and Challenges. In: System Development. Zupancic, J., Wojtkowski, W., Wojtkowski, W. G. and Wrycza, S. (eds.), Kluwer Academic/Plenum Publishers, New York, pp. 19-31.

Appendix A: Research publications

Nr	Title	Published
1	Projects as Learning Environments A Case Study in IS Education (Junker, T. and Omland, H.O., 2000).	Sein, M.K et al. (eds): Contemporary Trends in Systems Development. Kluwer Academin/Plenum Publishers, pp. 419 – 432.
2	Course Integration as Learning Environment for Increasing Competence (Nordheim, S. and Omland, H.O., 2002)	IS2002 Informing Science + IT Education Conference, June 19-21, 2002 Cork, Ireland (Available at: http://proceedings.informingscience.org/IS2002 Proceedings/papers/Nordh144Cours.pdf
3	Relationships between Developers' Competence, Methods and Practice in Information Systems Development: A Case Study (Omland, H.O., 2004).	Vasilecas, O. et al. (eds): Proceedings of the Thirteenth International Conference of Information Systems Development: Advances in theory, Practice and Education, ISD'2004, Vilnius, Sept. 9. – 11. 2004, pp. 305 – 316.
4	The relationships between competence, methods, and practice in information systems development (Omland, H.O., 2009)	Scandinavian Journal of Information Systems, 2009, 21(2), pp. 3-26
5	Actors' Competencies or Methods? A Case Study of Successful Information Systems Development (Omland, H.O. and Nielsen, P.A., 2009)	Proceedings of 20th Australasian Conference on Information Systems, 2 – 4 December 2009, Melbourne, pp. 215 – 224. (Available at: http://infotech.monash.edu/about/news/ conferences/acis09/Proceedings/pdf/022.pdf

PROJECTS AS LEARNING ENVIRONMENTS

A Case Study in IS Education

Hans Olav Omland[1] and Tore Junker[2]

Department of Information Systems
Agder University College
Kristiansand, Norway
E-mail: Hans.O.Omland@hia.no
Department of Information Systems
Agder University College
Kristiansand, Norway
E-mail: Hans.O.Omland@hia.no

INTRODUCTION

For a System Analyst educator, the following question may include most of the main educational issues:

> *What factors are most effective in creating a learning environment where students can learn to become System Analysts (SA) with knowledge, attitudes and identity that enable them to develop high quality Information Systems?*

This is a very wide question. In this paper we want to narrow it down by concentrating on one aspect of an answer:

> *What learning environments will provide opportunities for the System Analysis students to develop a reflective attitude and become aware of their own formative context?*

Even this is a very ambitious question to answer. We are looking for fragments to an answer. In this paper we discuss the theoretical background for a project course where we have applied some of the principles we discuss. We describe the components of the course, resources used and some initial impressions of learning environments.

SYSTEM ANALYSTS – ROLES AND COMPETENCIES

Turning to textbooks in System Analysis and Design searching for descriptions of a SA, we often find one or two pages with fairly standard text, listing typical demands

Contemporary Trends in Systems development,
Edited by Sein *et al.,* Kluwer Academic/Plenum Publisher, 2001

Table 1. Expected demands on a SA

Roles	Knowledge/experience	Personal qualifications
Consultant	Being a problem solver	Self-disciplined and self-motivated
Supporting expert	Being a communicator	Creative and imaginative
Change agent	Having analytic abilities	Able to work in environments that has considerable ambiguity and uncertainty
	Having experience in programming	

of a SA (Hawryszkiewycz, 1994) and (Kendall and Kendall, 1998) exemplified in Table 1.

Having all these qualifications, the SA should be able to master everyday tasks in developing systems. The competence of mastering may be described as a wheel of competencies (Fig. 1) (Borgen).

All these competencies are necessary to master the system development process. Depending on the actual task, the optimal mix of these competencies may be different, i.e. in a situation where human communication is important, the social competence is more important. In other situations technical competencies may be more important. The SA has to work with different actors demanding unique mixes of competencies.

Textbooks focus often on the professional/technical competencies. If the education of SAs is based on these textbooks, the SAs will be good at applying methods, techniques etc. to fairly accurate well defined theoretical and practical problems.

Unfortunately the world is more complex. Ciborra (Ciborra, 1998) states that using a model to describe the world, requires knowledge of the world, or the situation, in order to understand and interpret the model. If we try to construct the world solely from a model we will not succeed. The consequences in the IS field is that the SA must know, at least to a certain degree, the actual organisation in order to apply models for developing the IS. To obtain knowledge and other pieces of information about the organisation, the people working there, routines etc., SA needs to communicate with an open mind focusing on people, not his methods, at least not in the early phases of the development process.

Figure 1. Competence of mastering.

If education of System Analysts concentrates mainly on methodologies, methods and other technical aspects omitting interpersonal relations, communication, attitudes, reflections etc., the training will prepare the System analyst for a world that is very different from the one he/she will encounter in work situations.

Since the SA has to interact with many different people and in many different situations, the ability to communicate and co-operate and the ability to reflect on one's own actions is important.

In textbooks describing the role of the SA, this ability to reflect is seldom mentioned. Reflection before, under and after one's own action and experiences is of vital importance and enables learning and change of own practice.

Therefore this paper argues that the education of SAs must include subjects and educational approaches that expose the students both to mere technical and methodical issues and provide opportunities for them to become self-reflecting, good listeners and communicators, i.e. obtain the competence of mastering. In our opinion technical knowledge is necessary, but not sufficient to be a successful SA. Technical/professional knowledge is often the easier part of the learning process. Reflecting over actions requires time, effort and ability and may be painful.

This leads us to the meta concept of learning: "Learning to learn" as a deeper principle than just to learn to perform a given task. Learning to analyse, define and solve fuzzy problems will be a better preparation for future work situations. Since the SAs also are actors in the work situations, they need to reflect on themselves and their ways of solving problems. They may even themselves be the cause of the problem (Argyris, 1991).

THEORIES OF LEARNING

Learning is successful when it results in permanent changes of action. Is it possible to find and implement a good learning environment for SA students? We will present two theories, the theories of action and the learning triangle.

Theories of action

All human action is based on theories of action (Argyris and Schon, 1974). Argyris and Schon argue than one can differentiate between espoused theories of action and theories-in-use. Espoused theories of action are the theories people report as basis for the actions. Theories-in-use are the theories of action inferred from observations of people's actual behaviour. Most individuals are able to detect discrepancies between espoused and the theories-in-use of others, but are less able to detect similar discrepancies in themselves (Argyris and Schon, 1974). Reflection and double loop learning may be one way to overcome this problem.

Single and double loop learning. Single loop learning leads to more effective work routines. People improve their routines. The new knowledge is mainly within the domain of the established thought pattern that originally was used when the routines were established.

The double loop learning will influence the "formative context" of a person. This formative context can be described in terms of frames, thought patterns, institutional knowledge etc. Changes in the formative context will influence how people establish routines and give meaning to these routines.

Single loop learning is fairly easy, while double loop learning is difficult, painful and scary (Argyris and Schon, 1974). The reason is that double loop learning may lead to new perspectives on people, organisations and a new world-view. The Chinese expression for crisis illustrates this. It consists of two characters, carrying both the meaning of danger and opportunity. Double loop learning includes both these aspects and can be quite scary. The outcome of this kind of learning will not necessarily be positive.

Ciborra and Lanzara (Ciborra and Lanzara, 1994) report that people are under the influence of a pervasive and deep-seated texture of relations, a formative context, when skilfully executing their daily routines. This formative context is often the basis of their skill, and they are unaware of their actual practices in their daily work. Ciborra and Lanzara also state that systems and routines should be designed to help organisational actors perform a perpetual activity of reflection and self-questioning. These abilities are necessary to learn from experiences. How can SAs become more aware of their own formative context? Reflection is probably the most important factor.

The Learning Triangle

Learning i.e. permanent changed behaviour, involves three aspect stated in the learning triangle: knowledge, affections and action (Fig 2.) (Rørvik, 1994).

Learning is a lifelong process since we change all the time. Learning is most effective if the whole person is involved. In order to change some actions, a person must know other alternatives and details about them. Affections may be an important motivational force in the learning process. Action, "doing the stuff" is necessary to obtain experience.

In ordinary classroom education the knowledge part is normally emphasised. Learning methods and techniques that require action/practice are normally included in the curriculum of the SA. In the textbooks affections are not regarded as an important part of SA's education. This may limit SA's ability to communicate with users of IS, especially in situations where affections are encountered. Reflecting on experiences in real world situations will therefore make the students better prepared for their task as SA. Can students reflect in a classroom? Of course they can. They can also act in a classroom and gain experience. Still we argue that it is, in most cases, more important to reflect on experiences gained in "real-world" systems development situations.

Consequences for the Educational Approach

Optimal learning environments will consist of elements where students can exercise and experience all the three factors in the learning triangle. Students are different and learn in different ways. They will react differently in the same learning environments.

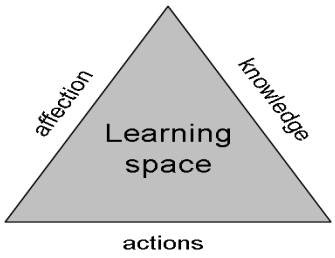

Figure 2. The Learning triangle.

It may therefore be a challenge to establish learning environments that provide good learning environments for them.

EDUCATIONAL APPROACHES

Developing high quality IS requires good communications between different participants in the development process. Many factors may hinder good communication like the truths we often are taught:

- *Be rational*	- *Be in control*
- *It is a question of win/lose*	- *Do not hurt peoples feelings*

Argyris and Schon (Argyris and Schon, 1974) argue that these ways of acting result in no or very little learning in action. Learning in action tends to evade us if we are occupied with controlling the situation, the conversation, or the people engaging us. Focus on control will make reflection difficult since reflection and learning always carries an element of uncertainty and risk.

Double Loop Learning Prerequisites

Is it possible to experience double loop learning directly without any practical experiences? We may theoretically reflect on our formative context or our tacit knowledge, but most of that reflection will probably be of the "espoused" theory type. When we act, we may become aware of our formative context or at least elements of it.

How do we establish a learning environment for double loop learning? According to the learning triangle we need to involve knowledge, affections and action. The aspect of affections is probably the most difficult and scary to include in a learning environment. Real-world project will, in our opinion, include all the three aspects of the learning triangle and provide an opportunity for the students to reflect on their formative context. We believe that activities leading to frustrations or a feeling of helplessness, may provide opportunities for reflection especially if the students are in a supportive environment where these reactions are not only accepted, but also considered as a normal part of the learning process.

Learning Communication

What is the best context to learn communication? Context is recognised as more and more important in understanding relationships between people and computers. Croon (Croon, 1998) argues that contextual approaches to people working with computers reveal the importance of understanding more about the character of social influences on technology beyond the context of use and design. Context is important in a learning situation since it gives awareness of problems and specialities in the given field. Through the "art" of awareness and reflection the student may be able to transfer knowledge from one context to another.

Major Educational Approaches

There are at least three major educational approaches for obtaining knowledge, skills and experiences. We believe that all these approaches are important and necessary in a good learning environment.

Text Book. Knowledge is often obtained by reading books. Books provide students with a basic overview and understanding of a given field, but do not give them any practical experiences. Reading will probably lead to more "head knowledge", but will not necessarily change their behaviour. The learning will constitute the knowledge leg of the learning triangle, but will lack the affective and action oriented dimensions. It may easily become a theoretical construct.

Cases. Students in the IS field are often applying theories and methods to practical problems in cases or assignments. Cases are defined by the teachers or are descriptions of situations or organisations in the "real" world. Cases are useful tools for giving students experiences in analysing and solving problems. These cases may form the background for developing IS. Using cases it is possible to supply information at any given time thereby making up for low quality work in earlier phases of the development process. This is normally not the case in real-world projects where students may have to "suffer" from consequences of poor work in earlier phases of the development process. This experience may be important for double loop learning, proving opportunities for training communication and vivid experiences for reflection.

Real World. Projects in the "real" world will give students experiences in meeting all kinds of people, problems, dynamic organisations, creeping requirements and their own frustrations. As far as we know, this type of projects is rare in the SA education. Since the real-world approach in our opinion is of great importance, it is the focus of this paper. This educational approach will include all the three factors of the learning triangle and will give the students opportunities to experience all aspects of the competence mastering.

We have discussed learning theory and some major implications on the learning theories. How can these theories be transformed into practical learning environments? In the following we will report on a case study where we describe some of the principles we implemented in the learning environment.

CASE STUDY

Background

The case study reports on the planning, implementation and evaluation of a partly project-based course for 4th semester undergraduate students in the information system (IS) and business administration (BA) streams at Agder University College. The course represents 40 % of a normal semester workload for a student.

Projects have been used in this course for several years, and the learning value of projects has more or less been taken for granted. In later implementations we have focused more specifically on the learning environment and the learning outcome of the course. Our main goal has been to create a realistic learning environment that would stimulate action and reflection and enable double loop learning. The case study is based on the 1999 implementation of the course.

Basic Implementation Principles

A project-based course may be implemented in many different ways, and the implementation will determine important characteristics of the resulting learning environment. Our implementation was based on the following principles:

a) Unique, Real World, System Development Projects. Only real-world projects introduce a realistic level of uncertainty and ambiguity. Hopefully, such projects will encourage student reflection on system development theory and methods and the actions involved in their application in this real-world setting. A realistic relation to the employing organisation can only be achieved when each project group has its own unique project.

b) Autonomous Project Groups. To stimulate student reflection, we believe that a high level of project group autonomy is crucial, including both the freedom to make their own decisions and the full responsibility to do so. By avoiding instructor approval of subtasks or milestones, each project group must continuously decide on when and how to proceed. This may be difficult without a certain level of reflection.

c) A Supportive Project Environment Most of the students involved, are usually inexperienced SAs, and most of them may never have experienced this kind of project-based learning. The projects groups therefore need support both in the system development and in the learning process. Different support approaches will affect these processes.

Solution-oriented support may be most efficient from a short-term, system development point of view, while reflection-oriented support may be better for long-term learning. This conflict is typical for real-world projects. In our experience, most employing organisations both accept and support the learning aspects of the projects, but naturally they will prioritise the application and therefore solution-oriented support. Most students also favour this kind of support, at least in the early phases of the projects. When they get accustomed to reflection-oriented support, they may have a more balanced view on different support approaches.

Instructors must consider both system development and learning aspects. Compromise may be the only viable solution. Even if reflection-oriented may be best for learning purposes, solution-oriented support may be necessary to achieve application goals. Single loop learning may be the outcome, but fortunately this kind of learning has its own value.

In our opinion, reflection-oriented support should be the main support approach. When solution-oriented support is preferable or needed, instructor should at least propose or mention alternative options without revealing their own priorities.

To make the projects more realistic, the available support should be limited, so that each project group has to decide on the optimal use of the assigned support resources.

The supportive environment's role as a "safety net" for the project groups may also stimulate learning. The mere existence of a supportive environment may induce a more explorative student behaviour.

d) Some Compulsory Milestones. Some compulsory milestones may help students organise their projects into more manageable pieces, giving them some useful and realistic short-term goals. Using real-world projects, instructors definitely have a need for some control of project progress and direction. Compulsory milestones can be used to introduce the desired, minimum level of control, and they can also include a minimum level of feedback. This feedback may be both solution- and reflection-oriented depending on the project phase and the kind of problems.

In addition, a few compulsory milestones increase the realism of project management and may encourage the project groups to introduce management by milestones in their projects.

e) Grading of both Product and Process. Since the projects are part of a course, they have to be graded. We believe that the grading method will influence the priorities of the project groups and the way they work and thereby the learning outcome.

A grading based exclusively on the final product may reduce the focus on the development process even if the process from a learning point of view may be more important. By grading both product and process, we hoped to achieve a student focus on both of them.

The uniqueness of real-world projects may represent a special problem when grading. Each project is often very different from the others, and it may be difficult to compare them.

In our opinion, the grades should mainly be based on factors that the project groups are able to control. Even if several important variables may be uncontrollable for the project groups, they should be able to control the quality of the development process to a reasonable degree.

The Course Implementation

The actual implementation of a course defines important aspects of its learning environment and thereby the learning outcome. Our implementation was based on the discussed, basic implementation principles and our own experiences. However, the total number of students and the resources available for the course limited the number of realisable implementation options. The main organisational elements are discussed below.

The Main Organisational Elements

Project Groups. Group size is an important variable in project-based learning. It determines the maximum project workload and may also have important impacts on internal group organisation and student involvement. In addition, group size determines the total number of project groups and thereby much of the instructors and assistant teachers workloads.

Active student involvement was a prerequisite in our approach. Small groups stimulate involvement, but project groups had to be large enough to handle the available and suitable real-world projects.

Table 2. Implementation structure – an overview

Phase	Activity	Compulsory
pre-project phase (mid November to mid January)	comprehensive pre-course briefing focusing on the projects	
	project group formation search for employers and project ideas	
	project proposal development	
	formal project proposal approval	X
project phase (mid January to mid April)	approx. 2 ½ weeks of project work	
	1st meeting with steering committee	X
	approx. 2 ½ weeks of project work	
	2nd meeting with steering committee	X
	approx. 2 ½ weeks of project work	
	3rd meeting with steering committee	X
	approx. 2 ½ weeks of project work	
	handing in application and project report	X

A total of 76 students and 4-5 students per group resulted in 18 different project groups. The students were themselves responsible for the establishment of the project groups. Both for project and learning reasons we strongly recommended mixed IS and BA project groups. Unfortunately only 5 of the 18 project groups had both IS and BA members[1]. A 2/3 majority of IS students and minimal prior contact between the two student groups may explain this. Early discussion of group goals and agreement on a written set of group rules was also strongly recommended.

Projects. To fit the study programs, the projects had to be system development projects focusing on the analysis and design part of the development process. Project proposals with a main focus on programming and implementation aspects were not approved.

Expected, total student workload for each project was 5-600 hours depending on the group size.

For both autonomy and motivational reasons, each project group had to find its own employer and produce a project idea. Reduction in instructor workload was a valuable side effect of this rule. To introduce some control of project relevancy, workload and complexity, the project proposal had to be formally approved by the instructors to become a project.

Steering Committee and Steering Committee Meetings. To make the project organisation more realistic, a steering committee including an employer representative and the two instructors was established for each project group. Three steering committee meetings with their project group was the main task of these committees.

To give the students valuable meeting experience, the project groups both planned and chaired these meetings. Participation was compulsory for all the members of the project group.

Discussing the projects and especially their progress was the main purpose of the steering committee meetings. A short presentation of a written progress report and discussions of important problems were normally the main issues on the agenda. Each project group received both oral feedback during the meeting and a short written feedback after the meeting. Written progress report and written feedback were compulsory parts of the final project report to enable formal evaluation of important parts of the development process.

Supportive Project Environment. Access to up to 22 hours of normal support for each project group was an effect of project proposal approval. Assistant teachers were responsible for 12 of these hours, and implementation-oriented issues were their special domain. The assistant teacher support therefore had a solution-oriented flavour, but they were asked to use more reflection-oriented support whenever suitable. The remaining 10 hours were divided between the two authors. Reflection-oriented support was our main support approach.

In addition to this normal support, special support was available in the pre-project phase and for groups experiencing serious problems. This ad hoc support had no predefined limit, and it was offered on a need and demand basis.

Project report. Each project group had to write a comprehensive, process and problem-oriented project report. First of all this report allowed us to include the development process in our grading, thereby achieving more balanced grades.

[1] 10 pure IS groups, 3 pure BA groups and 5 mixed groups.

Secondly, we believe that this report requirement induced a stronger process focus in the project groups, and that this focus might stimulate learning. Hopefully, the necessity to document and defend the process increased the level of reflection during project work. The retrospective reflection forced by the report writing itself, may also have valuable learning impacts.

Evaluation of the Course Implementation

Background. Informal feedback from and our own experiences with several implementations of this course had given us valuable insight into and understanding of how project-based learning may be organised to optimise the learning environment. To supplement and validate our subjective impressions, a comprehensive post-project evaluation based on different questionnaires for project groups, employers and assistant teachers was implemented. This evaluation was initially intended for internal use, but is now used as a pilot study for further research. The main goal of these questionnaires was to answer questions like:

- *How do students and employers rate projects as learning environments?*
- *How do they rate real-world projects compared to school-defined projects?*
- *What are the optimal organisational approaches, and what are the learning impacts of the different organisational elements?*
- *What is the optimal balance between project group autonomy and instructor intervention?*

An evaluation based exclusively on single item measures and post-project questionnaires has its severe limitations, but may still contribute to at least some preliminary answers to these questions.

The surveys were carried out before the project grades were published to avoid their impact on the responses. To be able to compare the responses with other project group identifiable data and our own group observations, we asked the project groups and employer respondents to include the project group number in their responses. Anonymous response was offered as an option, but none of the respondents chose that alternative. Our request for group-identifiable responses may have influenced both the response rate and the answers. However, high response rates, complete responses and the answers themselves seem to exclude major impacts of non-anonymous responses (Table 3).

Measured by average project grades, the respondents seem to be fairly representative both for the employer and for the project group populations.

For many of the students and some of the employers this was their first experience with real projects. Their basis for comparison may therefore be weak.

Analysis of Responses. The questionnaires included many different questions. Some of them focused on internal course issues, while others had a more general perspective. In this section we present the responses to the most important general ones in an attempt to answer the principle questions mentioned above.

a) How do students and employers rate projects as learning environments?
We asked the project groups several questions to try to reveal their attitudes toward project-based learning and courses (Table 4).

Table 3. Populations and response rates

	Population	Number of responses
Project groups	18	15*
Employing organisations	18	15*
Assistant teachers	4	4

* For 12 of the project groups both the project group and their employer responded

Their reported attitudes may be dependent on their feeling of project success, and the very positive groups had a slightly better average grade than the groups with just a positive attitude. On the other hand, only three of the five best grades were given to the most positive project groups. All the pure BA groups reported very positive in spite of their somewhat weaker IS background. There seems to be a reasonable correlation between the general attitude towards projects and their opinion on the extent to which project-based courses should be used.

When asked about positive and negative aspects of project-based learning, most of the groups agreed on the aspects reported in Table 5.

Even if the students often find project-based courses demanding, difficult and sometimes frustrating, they still want more of such courses. Only 11 of the 15 responding project groups included work hour statistics in their project reports. Most of these groups reported a total number of working hours within the expected time frame of 5-600 hours. Recording problems may however have resulted in too low figures, and knowledge of the expected workload may also have influenced the numbers in workload reports.

The employers' attitudes and experiences towards project-based learning were also mainly positive (Table 6).

Another indicator of employer attitude may be their willingness to be involved in another project if they have a suitable project. As much as 13 of the responding employers were willing to do so.

Table 4. Attitudes toward project-based learning and courses

General attitude	very positive: 9 positive: 6 (none were neutral or negative)
Extent to which project-based courses should be included in study programs	to a large extent: 9 to a certain extent: 6 (none answered to a small extent or not at all)
Do you want more project-oriented courses than today?	yes: 10 may be: 5 (no one answered no or don't know)

Table 5. Student reported positive and negative aspects of project-based learning

Positive	Negative
- realism	- workload (also compared to credits)
- test theory on real world problems	- dependency on the rest of the project group
- future work relevance	- uncertainty and ambiguity
- an active way of learning	
- motivating and interesting	
- learn how to co-operate	

Table 6. Employer attitudes towards and experiences with project-based learning

How was your experience with project-based learning for this kind of students?	very good: 9 good: 5*	average: 0	bad: 1** very bad: 0
How do you rate the value of such projects for the future work of the students?	very high: 9 high: 4	medium: 2	low: 0 very low: 0

* Among the 5 were the two groups with the lowest grade, a group with internal problems and a very technology-oriented group
** This was a group with an average grade

b) How do they rate real-world projects compared to school-defined projects?
As much as 12 of the 15 responding project groups preferred real-world projects. The realism of such projects was their main argument. The remaining groups wanted either a mix or only school-defined projects. Their arguments were a need for project work training and fairness in grading.

c) What is the optimal balance between project group autonomy and instructor intervention?

Project group autonomy was important, both when projects and groups were established and during project work. When establishing project groups, 11 of the responding groups chose the easy and safe way. For them former and well-established study groups were the main bases for project group formation. Informal feedback confirmed this impression. Level of ambition and group member competencies were group establishment criteria just in a few cases.

Nine of the responding groups reported that students should be responsible for both group formation and project ideas. Learning impacts of such activities and freedom of choice concerning fellow group members were their main arguments. The remaining groups thought that co-operation with "strangers" might be valuable, and that mixed IS and BA groups should be the rule.

In our opinion, the balance between project group autonomy and instructor intervention is an important implementation variable. We therefore asked both project groups and employers about their opinion on this balance.

Students often found the instructor focus on reflection-oriented support frustrating and difficult to accept. Post-project feedback on support approach seemed to be more balanced. In our experience most students want the autonomy, but dislike some of its consequences.

Table 7. Balance between group autonomy and instructor intervention

Project group opinion on the division of authority and responsibility between project groups and instructors	8 answered suitable 3 wanted a bit less intervention * 4 wanted a bit more intervention **
Employer opinion on instructor intervention	11 answered suitable 4 wanted a bit more intervention ***

* 2 of these groups were very technology-oriented.
** 2 of these groups had problems defining and delimiting their projects.
*** One of these groups had severe internal problem and one was not especially user oriented.

d) What are the optimal organisational approaches to project-based courses, and what are the learning impacts of different organisational elements?

These questions may be very difficult to answer, and we definitely need more data to answer them properly. Our impressions below are based on the respondents' opinions on some of the organisational elements.

Compulsory project proposal approval was introduced both as an important milestone and as a way of controlling project suitability, complexity and workload. Proposals had to be approved within the first 14 days of the semester. Project groups failing to fulfil this requirement, were not admitted into the course. The instructors assisted and advised the project groups in their project proposal work, stimulating action and reflection. Most project groups had to hand in at least three successive and gradually refined project proposals before the project proposal finally was approved. All the responding groups reported a need for project proposal approval. Reasonable workload and complexity were their main arguments, but some also mentioned the value of the proposal work itself. 10 of the employers answered that project proposal approval was very desirable. The employers' main arguments were approval impact on project work.

The three meetings with the steering committee were introduced both for milestone and control purposes. As many as 13 of the responding groups wanted such meetings (or something similar) during the project phase. Their importance for and impact on project work were the typical arguments. These groups also reported that 3 meetings seemed to be the optimal number. Most of the employers agreed on the importance of the meetings, and their usefulness for both project groups and employers. Employers experienced these meetings as valuable arenas both for learning, creativity, discussions and decisions.

Resource Requirements, Resource Availability and Workload

Our implementation of this project-based course required more resources than traditional, comparable courses. Additional instructor and assistant teacher workload was the main reason. Student credits are the normal basis for faculty teaching load calculation at our college. The normal workload of a full time student is 10 credits per semester. For faculty, the normal teaching load is 4 student credits.

Our course represents 4 student credits, but the 1999 implementation was calculated to 6 instructor credits because of the additional instructor workload involved. The use of assistant teachers may differ, but we often use 200-250 assistant teacher hours for comparable courses. We used about 320 hours for our course in 1999.

Table 8. Estimated instructor workload per project group

Instructor activity	No of hours per project group
Normal, maximum support (incl. necessary preparations)	10*
Estimated average ad hoc support	3
Steering committee meeting (incl. preparations and feedback)	21**
TOTAL	34

* The real, normal support for each group differed, but 10 hours was the maximum.

** Provided that both instructors attended all meetings. This was desirable, but not always possible.

Instructor workload per project group may be another interesting resource requirement measure. Unfortunately, the exact hours spent on support and meeting were not registered. The figures below are therefore our best estimates.

A 2/3 majority of the responding project groups reported that the level of available resources was reasonable, while most of the other groups reported small additional needs.

Software tools are another important resource in software development projects. The project groups had a free choice of such tools, but the assistant teachers supported only a defined selection of them. In addition, the college bought no new tools just for project use.

CONCLUSION

The responses from both project groups and employers were in general very positive. Together with our own observations they definitely strengthened our belief in real-world projects as learning environments for future System Analysts. The surveys and our subjective opinions are however preliminary, and cannot represent conclusive evidence of learning outcome. Considering all the hard work in the projects, the responses indicate valuable experiences for the students involved. Depending on the implementations, some of these experiences may be very different from those of more traditional courses.

In addition, a closer and more active relationship between IS students, IS educators and IS practitioners may be a valuable side effect of the real-world project approach.

More research is definitely needed both to develop measurement methods for the learning outcome and to define optimal organisational approaches. It may be hard to establish the causalities between organisational elements and learning. In addition, real-world projects tend to increase the number of variables that are difficult or impossible to control. There is also a need for a closer integration between learning theories, methods for estimating learning needs and different learning approaches. Co-operation between institutions applying the project approach may also be very desirable.

In our ongoing research in this area, we are searching for indications of double loop learning by measuring changes in important professional attitudes. In addition, we would like to establish the total and relative importance of different organisational elements of our approach and some of their impact on learning.

REFERENCES

Argyris, C. and Schon, D. “Theory in practice: Increasing professional effectiveness”. Jossey Bass, 1974.

Argyris, C. “Teaching Smart People How to Learn” *Harvard Business Review,* 1991

Borgen, Ø. Adopted from an unpublished model, (Agder University College)

Ciborra, C. “Crisis and foundations: an inquiry into the nature and limits of models and methods in the information systems discipline”, *Journal of Strategic Information Systems,* 7, 1998.

Ciborra, C. and Lanzara, G:F: “Formative contexts and information technology: Understanding the dynamics of innovation in organisations”, *Journal of Accounting, Management and Information Technology*, Vol. 4 No 2., 1994

Croon, A “Reframing the Notion of Context in Information Systems Research”, *Proceedings of the 21. Information-systems Research seminar in Scandinavia,* volume 1, 1998.

Hawryszkiewycz, I.T., “Introduction to System analysis and Design”, Third Edition, Prentice Hall, 1994.

Kendall, K.E. and Kendall, J.E. “System Analysis and Design”, Fourth Edition, Prentice Hall, 1998

Rørvik, H. “Læring og utvikling – Det pedagogiske oppdraget”, Universitetsforlaget, 1994.

Informing Science **InSITE - "Where Parallels Intersect"** **June 2002**

Course Integration as Learning Environment for Increasing Competence

Stig Nordheim and Hans Olav Omland
Agder University College, Kristiansand, Norway

Stig.Nordheim@hia.no Hans.O.Omland@hia.no

Abstract

Creating a realistic learning environment for Systems Analyst and Design (SAD) students is a challenge. Integrating discipline-oriented courses is one way of creating such an environment. This paper discusses a possible integration between two courses where the courses provide a basis for experiences that enrich the learning environment and give the students important competence in the IS field. The paper describes levels of integration and some consequences for students and lecturers. The discussion ends in a proposal for integration of the two courses. The integration of the courses will be on a so-called temporal co-ordination level where the main contributory factors are deliverables and the co-ordination of time allotted to the courses. The question of course integration is seen both from the lecturers' perspective and the students' perspective. The integration was tried out in the spring term 2002.

Keywords: Competence, interdisciplinarity, learning environments, course integration

Introduction

The final year of the IT and information systems (IS) undergraduate study at Agder University College (AUC) includes a 2.5 credit course in project work and quality assurance (QA), and a 5 credit project based course in application development. In this paper we sketch a possible integration of these two parallel courses, creating a learning environment based on the simultaneity and complementary objectives and content of the two courses.

We aim at answering the following question: How can an integration of the two courses provide a better learning environment for the IS students?

Students that attend the two courses are educated to become Systems Analysts and Designers (SADs). The courses are designed to contribute to building their competence in developing information systems (IS).

In the paper we discuss course objectives, include a competence perspective, present concepts of interdisciplinarity and integration, and propose a feasible integration of the two courses. The integration is based on the competence and integration models presented.

Course Objectives

The integration is based on the learning objectives of the two courses. In the following we present the objectives of the courses and discuss the integration of them. We discuss the possibilities of reaching these objectives at the end of the paper.

IS-3000 Project - Application Development

The course aims at

- providing an overall understanding as well as practical experience in computer-based information systems development
- providing students with an opportunity to apply, integrate and reflect on acquired skills in the course
- developing professional communication and collaboration skills

IS-3200: Project Work and Quality Assurance

This course introduces

- the significance and function of quality assurance (QA). A comprehensive understanding of quality is emphasised
- an efficient project management (PM) which is essential to control and co-ordinate the different development and QA activities in a systems development project
- different theories, principles and techniques of the QA field

Compatibility of the Course Objectives

The objectives of the two courses are complementary. There is no inherent contradiction to dissuade us from attempting course integration. And there are arguments for integration, to be presented later.

The IS-3000 course is a practical project in developing IS. The course integrates knowledge that the students have acquired earlier in the study programme. A practical project is a common way of integrating different subjects.

As for IS-3000 the IS-3200 course expands the realistic practical experience in developing computer-based IS, by providing guidelines for PM and QA.

IS-3000 provides an arena for practising theories learnt in IS-3200 as the latter course aims at securing quality in the development process and in the finished product. The students can to some extent test the theories they learn in IS-3200 on their work in IS-3000.

The following topics are related to both courses:

- Project management (estimates, planning, tracking)
- The systems development process (analysis, requirements, controlling changes, inspections, testing, process improvement)

The following topics are only related to IS-3200:

- Overview of QA methods (TQM, ISO 9001, CMM)
- Vendor relationships
- Configuration management

The following topics are only related to IS-3000:

- SW development tools

For a more complete description of the interface between the two courses, we will later present a list of deliverables (Table 2). The courses are compatible and their objectives may benefit from integration. When presented with an outline of an integration of the two courses, former students felt that integration would benefit both courses. A possible disadvantage may be that practical projects consume time. The students have to cope with the nitty-gritty details of the projects. That may "steal" time from the more theoretical learning process. A limitation is the small scale of the projects, 1100-1400 hours.

Course Objectives and the IS Field

Dahlbom and Mathiassen (1997) discuss the computer science profession. One part of their conclusion reads as follows: "Over the years, the focus of our profession has shifted from numerical analysis to programming to software engineering to human-computer interaction to networking." This expresses that the profession is arriving at a more holistic view consisting of an artefact focus, a culture focus, and a power focus (Dahlbom and Mathiassen, 1997).

Dahlbom and Mathiassen are discussing the computer science profession, we are discussing the IS field. Use of IT is spreading to every trade and used for many different purposes. That should call for a more holistic perspective on developing IS. At the same time different parts of an IS get more and more complex requiring specialists that know these parts in detail and thereby may loos the holistic perspective. By integrating courses we hope to move our students towards a more holistic perspective on IS development, with a practical project being a substantial means for learning.

The SAD will need different skills to master the daily tasks in ever changing environments. That does not necessarily imply that every SAD need the same skills, but it certainly implies that any given group of SADs need to master the competencies necessary to develop successful information systems. Both courses' objectives contribute to this goal.

Let us take a closer look at competence and how to obtain it.

Competence

Competence as a concept is demanding to define. We will therefore attempt to approach competence from different perspectives. The first entry under competence in Webster (1993) reads: "a sufficient supply; SUFFICIENCY". The Concise Oxford Dictionary (Sykes, 1982) defines competence as "ability to do" something or "ability for a task". Competence has a practical dimension in that it gives the holder sufficient ability to carry out something.

Figure 1: Competence of mastering

Competence will always be used in a context. Some types of competence are more general and may be valid and useful in different contexts. Developing IS calls for a competence of mastering, that a person/organisation have sufficient competence to master the daily tasks needed for systems development. The competence of mastering is a generic competence that consists of other competencies (Borgen), Figure 1.

All these competencies are necessary to master the systems development process. Depending on the actual task, the optimal mix of these competencies may differ, i.e. in a situation where human communication is important, the social competence is more important. In other situa-

tions technical competencies may be more important. The SA has to work with different actors demanding unique mixes of competencies.

The world is complex. We create models to understand the world. Ciborra (1998) states that using a model to describe the world requires knowledge of the world, or the situation, in order to understand and interpret the model. The consequences in the IS field is that the SA must know, at least to a certain degree, the actual organisation in order to apply models for developing the IS. To obtain knowledge about the organisation, the people working there, routines etc., the SA needs to communicate with an open mind focusing on people, instead of his methods or tools, especially in the early phases of the development process. In such situations it is important to keep both artefacts, culture and power focus.

This calls for organisational competence. Organisational competence is learnt in an organisational learning environment and may imply interdisciplinarity. Our understanding of IS follows Checkland and Holwell (1998), regarding IS as IT used in a context. The context is normally an organisation or an organisational unit, and is represented by the users of the application (Dahlbom and Mathiassen, 1997).

We assume IS competence to be a kind of integrated or interdisciplinary competence. Hager and Gonczi (1996), argue that the concept of integrated competence may be conceptualised as interpersonal skills, cognitive skills, affective attributes and technical skills in the context of professional tasks. This also fits with the view of user competence as the user's potential to apply technology to its fullest possible extent (Marcolin et.al., 2000).

To emphasise IS competence as opposed to IT competence can be said to conform to the more recent understandings of computer literacy. According to Gripenberg (1998), the concept of computer literacy has moved from focusing on special knowledge domains with pre-specified levels of competence, to a more functional view that focus more on if and how well people can perform with the computer in a context where they can solve their primary tasks. Dahlbom and Mathiassen (1997) also support this change of focus.

The competence model presented is a general competence model. When it is used in the IS field and filled with specifics of IS, it will be contextualised and contain more IS specific competence.

We believe that projects are good learning environments for students to build competence. The IS field is complex and diverse. Different actors exercise power in the process and may have their own agenda in the development of the IS. These agendas may be different from the main agenda for the development project, if such agenda exists. These differences may surface in an actual setting and provide an important aspect of the learning environment.

Integration of the two courses in this case provides at least a more complex environment for the students. The objective of integration is to provide a learning environment where the students can get competencies that are not obtainable by running the courses separately. Competence in integrating disciplines is one example. Such integration will also provide a rich environment for reflection.

Before we discuss the practicalities of the integration between the two courses it is necessary to discuss and decide on the kind of integration that is possible and acceptable.

Course Integration

Courses or subjects are integrated to reach some objectives. These objectives will influence the type of integration and may involve two basic forms of integration, namely multidisiplinarity and interdisciplinarity. We will discuss these two forms in relation to our courses.

Discipline may be defined as a branch of learning or a field of study characterised by a body of intersubjectively acceptable knowledge, pertaining to a well-defined realm of entities (Kockelmans, 1979). In our

case the two courses represent partially overlapping subjects consisting of features from many distinct disciplines.

Interdisciplinarity may be seen as bringing together distinctive components of two or more disciplines (Nissani, 1997). The term interdisciplinary implies a synthesis or integration, so we may also use the term integrated. Klein (1990) uses the two terms interchangeably.

The term interdisciplinary needs to be distinguished from multidisciplinary. Kockelmans (1979) describes multidisciplinary as related to more than one discipline, although there may be no connection at all between the disciplines involved. Multidisciplinarity can be seen as a juxtaposition of established disciplines, it is essentially additive and not integrative (Klein, 1990). These concepts are not to be confused with transdisciplinarity, a quest for unified knowledge that is outside our focus of interest.

Two alternative ways of illustrating interdisciplinarity is Figure 2 (inspired by Klein, 1990) and Figure 3 (from Harden, 2000). Both emphasise the integration of competencies (A, B, and C) that leads to new competence (D).

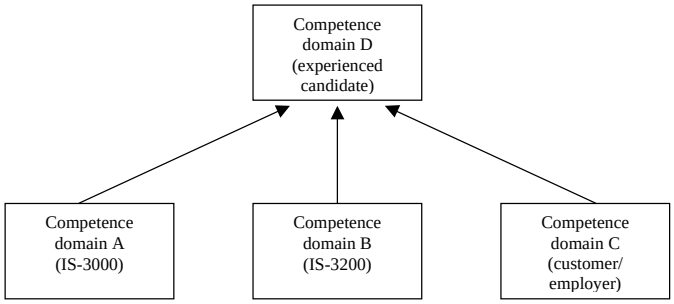

Figure 2. An illustration of interdisciplinary competence

According to Nissani (1995) one may describe interdisciplinarity by weighing 4 variables: the number of disciplines, the distance between them, the degree of novelty, and the degree of integration.

If we look at the IS profession itself, it is by nature interdisciplinary. The study of Lee and Trauth (1995) shows that to be an IS professional demands knowledge and skills in technology, management and interpersonal relations. Their study shows that this requires co-operative efforts and multidisciplinary or even interdisciplinary approaches to IS education. In their own words:

> "our results also indicate that IS graduates will require both more breath and depth of
>
> education across the dimensions of technology, business, and human relations".

We can easily relate these three dimensions to our competence domains in Figure 2 and 3.

Systems development projects imply at least multidisciplinarity. We may regard IT development project teams as largely multidisciplinary, involving team members from several disciplines. But the IT system itself may be re-

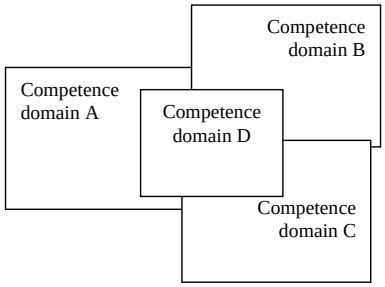

Figure 3. An alternative view of interdisciplinary competence

garded as interdisciplinary; it represents an integration that is more than the sum of its parts.

The course integration planned in this case, will hopefully imply that a project (IS-3000) with better process and product quality is carried out, and that the practical consequences of QA (IS-3200) is experienced in real life. This could hardly be accomplished if the two courses were run in isolation.

Integration: A Balance between Ambition and Benefits

A finer distinction than the multi/interdisciplinary distinction given previously is found in Harden's (2000) article. From a medical education viewpoint he formulates what he calls an integration ladder. This ladder consists of 11 steps moving from discipline based to integrated teaching and learning (See figure 4).

In the first 4 steps (isolation, awareness, harmonisation, nesting), the emphasis is on the distinct subjects or disciplines. Moving up the integration ladder, the last 7 steps emphasise integration across several disciplines (temporal co-ordination, sharing, correlation, complementary, multidisciplinary, interdisciplinary, and transdisciplinary). This fine-grained taxonomy proposed by Harden (2000) may be useful for classifying and planning IT/IS competence development in itself, as well as developing the application domain competence that is so crucial to business. Nevertheless we may well have in mind that Berger (Apostel, 1972) warns that hierarchies are ill advised in the absence of well-developed theory.

Using Harden's (2000) integration ladder as a curriculum-planning tool, we will discuss feasible integration options (see Figure 4) and the possible integrated competence the students may get.

Beginning from the bottom of the ladder, isolation (step 1) would mean that each of the two courses was taught independently, paying no attention to the other course. In the case of awareness (step 2), one lecturer would be aware of the contents of the other course, and be able to avoid any overlap between the two courses. Moving up to harmonisation (step 3) would imply a communication and consultation about the two courses, adapting to each other's curricula. Nesting (step 4) the two courses means targeting skills related to the other course, so that teaching is related to the broader curriculum outcomes of the sum of the two courses.

At present we focus on a possible integration at step 5-7 and will therefore give more details on these steps. In temporal co-ordination (step 5) each course is responsible for it's own teaching. But a temporal co-ordination of the two courses would include adjusting the dates when different topics are taught. The students themselves are expected to uncover the relationships between the two parallel courses. Very often the so-called 'integrated teaching programmes' are in practice temporally co-ordinated programmes.

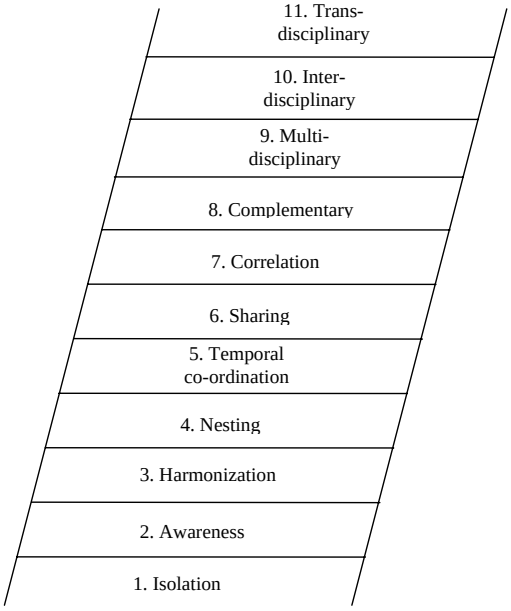

Figure 4. Harden's integration ladder

A sharing (step 6) would imply joint teaching in the two courses, with a focus on the shared concepts, skills and attitudes. Typically this would be a situation where two departments share parts of a common course, and the course is considered an end in itself. Moving further towards integration, we could try correlation (step 7). This means that emphasis would still be on the different courses, with discipline-based courses taking up most of the time. But in addition integrated teaching sessions would be introduced, to bring together areas of common interest to each of the courses.

Moving on to a higher degree of integration, a complementary programme (step 8) would have some subject-based teaching, but the common or integrated sessions would constitute the dominant part of the courses. The higher integration steps (9-11) were included in our preceding discussion of course integration concepts.

Our focus on the level or degree of course integration needs to be put in perspective. The degree of integration is the only independent variable we consider here, and we assume that it is important for the learning outcome. This is illustrated in Figure 5.

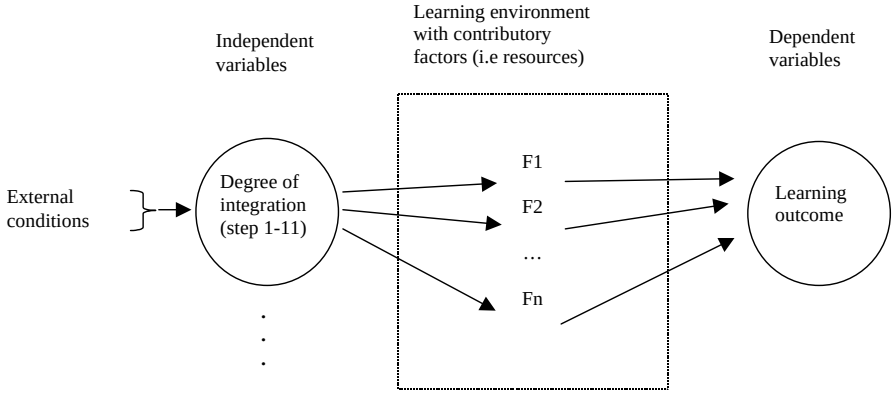

Figure 5. Relationship between degree of integration, resources and learning outcome

In the following we will discuss the most important contributing factors. Harden (2000) is mainly focusing on course integration from the lecturers' point of view. As we include the learning outcome in our framework we need to consider contributing factors for the students' part as well. Harden's ladder that we use from the lecturer's perspective has the contributing factor: parallel or concurrent teaching with co-ordination of similar elements in the different courses (step 5).

In IS-3000 there is no lecturing at all. In our integration we have therefore used deliverables like plans and reports to co-ordinate activities. These deliverables are designed to apply knowledge from one course to perform activities in the other course i.e. the quality plan to be delivered in IS-3200 is based upon the actual project in IS-3000.

In IS-3000 project supervision is a contributory factor for the lecturers' part. It is supposed to function as a practical help in the project and is carried out in a group setting. Students are prepared for the actual systems development through courses earlier in their study.

So far we have discussed the lecturer's perspective in the integration, i.e. course integration. The student's perspective is considered to be integration of knowledge and thereby increased competence. The following factors are considered as important for integration on the student's part:

- project
- integration of discipline knowledge
- use of different competencies
- experiences
- reflection
- receive supervision.

The basic learning environment is project work. The students are required to do projects that require integration of different discipline knowledge. These factors are therefore not regarded as course integration factors, but as knowledge integration factors. This is the basic reason for organising the IS-3000 course as a project. In this learning environment we find that integration for the student's part in fact may reach as high as step 8 in Harden's ladder. Even though Harden does not describe students' knowledge integration, we assume an improved learning outcome.

	Course integration (Lecturer part)	Knowledge integration (Students part)
Contributory factors	Parallel/concurrent teaching Co-ordination of deliverables Supervision of projects	Project Integrate discipline knowledge Use of different competencies Experiences Reflection Receive supervision
Level of integration	Step 5	Step 8

Table 1. Contributory factors of the learning environment

As lecturers we attempt to integrate the courses at step 5, temporal co-ordination. Due to the project in IS-3000 and the contributory factors we expect the integration for the students' part to be at a higher level than the lecturers' temporal co-ordination (step 5). Table 1 gives an overview of the integrating factors.

The contributory factors are important for the following affected interest groups: students, lecturers, potential employers and the college. In figure 6 we display the relationship between the students and lecturers as the two major interest groups, the contributory factors, and degree of integration.

The factors in table 1 are expected to contribute to increased competence. In the following we will discuss some of factors in the implementation of the courses. The details will be discussed later under the chapter: Plans for a pragmatic implementation.

Lecturers' Contributory Factors

We follow Harden's ladder and add supervision. Supervision is to some extent a substitute for the lecturing in IS-3000.

Students' Contributory Factors

The frame for the students' activities is the project. Within the project context the other activities will take place.

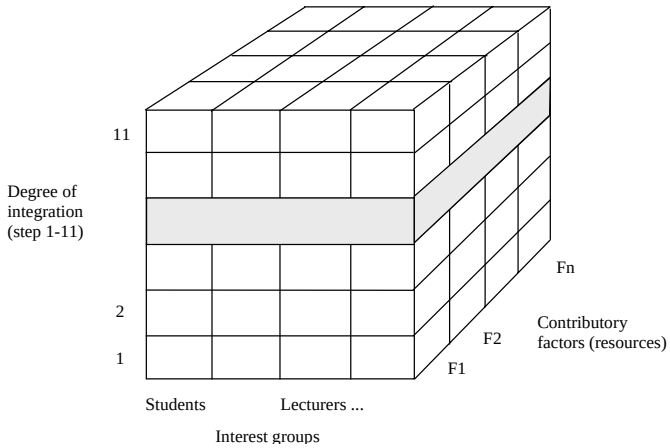

Figure 6. Interaction between interest groups and contributory factors

The students have to apply knowledge from different disciplines, both from IS-3200 and previous courses. This knowledge has to be integrated in the project and its context. The students have to master their development task and thereby utilise the four different types of competence, namely the social competence, professional competence, action competence and learning/change competence. Although it is difficult to establish a direct relationship between the use of these competencies and the end result, it is assumed that the quality of the result, i.e. the developed IS will depend on the use of these competencies. In our context the students have to write reports, present their work orally and have their work assessed. We assume that the assessment activities will reveal an increased competence. The students themselves will document some of the increase of competence as they are expected to reflect on the experiences they encounter during the project work.

The basic elements of our implementation are projects, group work, deliverables and assessment.

Projects

Projects are useful as interdisciplinary learning environments especially if they contain problem solving requiring different disciplines. In our case the project integrates the systems development exercise with the quality assurance discipline. The students have to practice quality assurance and are measured according to their competence to implement quality in their projects. In the IS-3000 course the projects are not approved before they reach a certain complexity. That means the students have to handle complexity. The introduction of the quality aspect increases complexity and thereby an experience not achieved if the courses were run separately.

Project groups give the students opportunities to practice the skills of communication both internally in the group and externally.

Deliverables

Some of the deliverables are common for the two courses. The students have to consider how to satisfy requirements from both courses and thereby get experiences in communication with focus on integration.

Some of the deliverables require a reflection that demands interdisciplinary thinking.

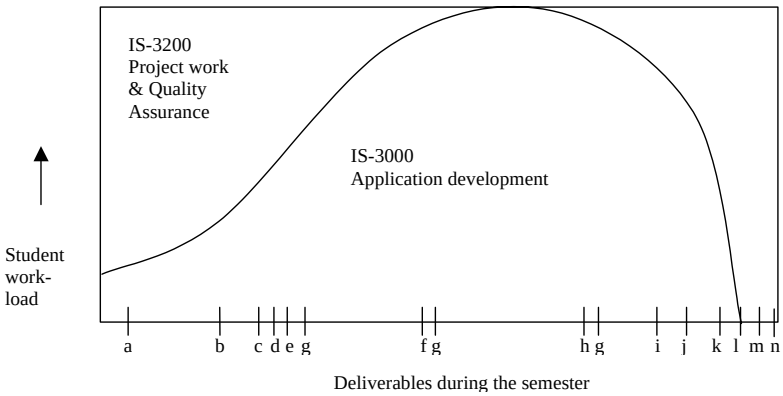

Figure 7. Suggested course integration with deliverables

Student Assessment

The ways the students are assessed will indicate what the lecturers deem important. Our experience is that students are eager to solve the technical problems. They are less competent in communication and designing their own development process to fit the actual context. We therefore emphasise communicative competence and the competence to reflect on the different problems the students meet during the project work. The students must integrate the quality assurance disciplines with the systems development exercise and be able to reflect on this integration. This reflection is recorded in a study report.

Provision of this learning environment can not guarantee increased competence. The actual increase in competence depends on the students' own efforts.

A Pragmatic Implementation

For the lecturers' part we integrate at the temporal co-ordination level. To move further up the integration ladder would require extensive and time-consuming syllabus changes. We will not discard further integration in the future, but we prefer a cautious and stepwise integration where the effects can be evaluated. This also fits well with the quality reform of higher education to be implemented in 2003.

Most of the contributory factors for the lecturers' part are integrated in the courses by co-ordinating deliverables and teaching as illustrated in Figure 7. The teaching in IS-3200 is adjusted to contribute to the progress of IS-3000.

Deliverables a-n are listed in table 2. In the beginning of the semester IS-3200 occupies much time for providing the theory for the QA aspects of the project. The theoretical knowledge for IS-3000 is provided in earlier semesters. Later in the semester more time is provided for the project work. After the project is finished and the deliverables in IS-3000 are all handed in, the students will have time to finish the required deliverables in IS-3200.

For the students' part the contributing factors may provide learning through the actual project work and production of reports. Plans/reports mentioned as items a, b, c, d, l, and m can only be produced successfully by integrating knowledge from the two courses and relying upon the experiences from the project work. Reports j, n, and i cannot be successfully completed without the students integrating experiences from the project with the theories they base their development on.

We will in the following discuss changes in the assessment and deliverables. Finally we will briefly mention other elements in the courses.

Assessment of Learning Outcome

To assess the learning we examine the students. In IS-3000 the assessment will be arranged as described below. The exam is inspired by practices at Aalborg University, Denmark.

Assessment will be based on written reports and an oral group exam. The student groups shall deliver the following written reports:

1. A systems development report that describes the development, the choices, successes and failures in the work they have done and the result of the work.
2. A study report that reflects on the work done in the project. The students have to choose some theories to reflect from and compare their experiences with.
3. A report describing the co-operation in the group. This report will also contain reflections of why the co-operation developed as it did.
4. A short individual evaluation of the other students in the group.

The students will participate in an oral group exam where they present the project and are examined both in the product they have produced and the reports they have written. This examination will include reflection on their work and presentation of different perspectives both on the reports and the product/application.

The assessment results in an individual grade and will be based on the written reports, the oral group ex-

Deliverables	IS-3000	IS-3200
a) Preliminary project plan	X	X
b) Quality plan I (without test plans)		X
c) Revised project plan	X	X
d) Quality plan II (test plans)		X
e) Analysis documentation	X	
f) Design documentation	X	
g) Steering committee documents	X	
h) Implementation documentation	X	
i) Systems development report	X	
j) Study report	X	
k) Group work experience report	X	
l) Quality evaluation report		X
m) Project tracking and evaluation report		X
n) Self and group evaluation for the exam	X	

Table 2. Deliverables

amination and the students` evaluation of themselves and each other.

These elements require the students to communicate and understand that good communication is dependent upon competencies given in the competence model. The exam itself will also be a learning situation where the students can practice the competence they have obtained and integrate the different subjects to form a holistic view of both their work, the result of their work, the users' needs and their own group process.

Required deliverables are described in table 2. These deliverables constitute the reports mentioned above.

The project management evaluation report includes reflecting on how successful the project (IS-3000) was, compared to the theory (IS-3200). Reflection and evaluation on how useful the theoretical quality (QA) and project (PM) subject matter (IS-3200) was for a successful real life project (IS-3000) are included in the study report in IS-3000.

Figure 8 illustrates a co-ordination of these deliverables based on time elapsed in the project.

Groups and Responsibilities

The project is carried out by groups of 4-6 students.

The students are responsible for finding the projects. Students are encouraged to utilise their network to find projects in ordinary companies. Hopefully they get access to project partners that are willing to use time on the project. The projects have to be approved by the lecturers assuring that the projects have the complexity and variety required.

The groups will receive supervision during the project work, both in the development of the system and in the group processes.

Groups provide an environment for communication and therefore opportunities for building competence in that field. It also provides a more complex environment that the students will have if they work individually. A group of 4 to 5 students has capacity to solve more complex problems contributing to more learning. The group organisation will give the students a feel for the need of organisational competence. The relative close co-operation with companies will also build some organisational competence.

Evaluation of the Course Integration

An ideal evaluation would include a control group, but this is not possible because of limited resources. Instead we will interview former students and use their retrospective reflection, and compare with evaluations from previous years with no course integration. After one semester we will evaluate whether course integration gives the expected benefits. In this semester the evaluation will mainly be based on the students self report and our evaluation of the courses.

Conclusion

The two courses described lend themselves to integration as the theoretical IS-3200 can use the IS-3000 project as a practical case, and IS-3000 will be even more realistic as the element of quality is integrated and followed up throughout the project. There are no conflicts in the objectives of the two courses in relation to the integration and the expected learning results. The hard work in the integration process will probably be the practical work of putting the plan into practice.

Students taking the courses can build and test their competencies in each of the two disciplines. The learning environment provides opportunities to integrate the two courses and experience the resulting opportunities and problems. These learning opportunities continue until the assessment is conducted.

Integration of the two courses provides, in our opinion, a better and more realistic learning environment for the students, and is more inspiring for the lecturers.

The final test of the integration and the creation of a better learning environment rest with the lecturers, the students and the implementation of the proposed plan. It can only be tried out and measured afterwards.

The feedback from the students on the integration will be input to a discussion whether further integration might provide an even better learning environment. Contrary to our expectations, the student feedback may even indicate a lower learning outcome of the integration than two isolated courses.

In future research it could be interesting to see whether teaching at step 5 can enable students to integrate knowledge at even higher steps, such as step 8 on the integration ladder. Testing the possible knowledge integration is another future challenge.

References

Apostel, L. et.al. (1972) *Interdisciplinarity: Problems of Teaching and Research in Universities*, Paris: Organization for Economic Co-operation and Development.

Borgen, Ø. Adopted from an unpublished model, (Agder University College)

Checkland, P. and Holwell, S. (1998). *Information, Systems and Information Systems – making sense of the field*, Chichester: John Wiley & Sons.

Ciborra, C. (1998). "Crisis and foundations: an inquiry into the nature and limits of models and methods in the information systems discipline", *Journal of Strategic Information Systems*, 7,1. 5-16.

Dahlbom, B. and Mathiassen, L. (1997). "The Future of Our Profession." *Communications of the ACM*, Vol 40, No 6.

Gripenberg, P., (1998). "*Computer Literacy: Still An Evolving Concept?*" Proceedings of the 21st Information Systems Research seminar in Scandinavia, IRIS 21, Aalborg, 253-263.

Hager, P. and Gonczi, A.. (1996). "What is competence?" *Medical Teacher*, Vol 18(1), 15-19.

Harden, R.M. (2000). "The integration ladder: a tool for curriculum planning and evaluation," *Medical Education*, Vol 34(7), 551-557.

Klein, J.T. (1990). *Interdisciplinarity: History, Theory, and Practice*, Detroit: Wayne State University Press.

Kockelmans, J.J.(Ed.). (1979). *Interdisciplinarity and Higher Education*, University Park: The Pennsylvania State University Press.

Lee, D.M.S. and Trauth, E.M. (1995). "Critical Skills and Knowledge Requirements of IS Professionals: A Joint Academy/Industry Investigation," *MIS Quarterly*, Vol 19(3), 313-345.

Marcolin, B.L., Compeau, D.R., Munro, M.C., Huff, S.L. (2000). "Assessing User Competence: Conceptualization and Measurement," *Information Systems Research*, Vol 11(1), 37-60.

Nissani, M. (1995). "Fruits, Salads, and Smoothies: A Working Definition of Interdisciplinarity," *Journal of Educational Thought*, Vol 29(2), 121-128.

Nissani, M. (1997). "Ten Cheers for Interdisciplinarity: The Case for Interdisciplinary Knowledge and Research," *Social Science Journal*, Vol 34(2), 201-217.

Shaft, T.M. and Vessey, I. (1998). "The Relevance of Application Domain Knowledge: Characterizing The Computer Program Comprehension Process," *Journal of Management Information Systems*, Vol 15(1), 51-79.

Sykes, J.B. (Ed.). (1982). *The Concise Oxford Dictionary of Current English*, 7th ed., Oxford: Clarendon Press.

Webster (1993). *Webster's Third New International Dictionary of the English Language unabridged*. Kønemann.

Biographies

Stig Nordheim (M.Sc.) has industrial and academic experience with different teaching methods. Current

research interests include IS competence.

Hans Olav Omland is researching competence needed for Systems Analysts and designers to design Information Systems. Current research interests include both IS professional competence itself and competence needed to use the IS competence.

RELATIONSHIPS BETWEEN DEVELOPERS' COMPETENCE, METHODS AND PRACTICE IN INFORMATION SYSTEMS DEVELOPMENT: A CASE STUDY

Hans Olav Omland*

1. INTRODUCTION

The purpose of this paper is to contribute to an understanding of the relationships between developers' competence, methods, and practice in information systems development (ISD). The paper is a case study of an actual ISD project.

Developing an information system is an activity where the actors use their competencies and apply methods in a given social context. Though developers' competence, methods, and practice are separate issues, they are very much related. The relationships are dynamic since the actors are interacting with each others and with the users of the information system when developing the system. The relationship is probably two-ways, i.e. the developers' competence, methods and practice will mutually influence each other.

There are many different ISD methods available today and they "differ greatly, often addressing different objectives" (Avison and Fitzgerald, 1988). There are also a number of implicit and explicit assumptions and views behind each method (Iivari and Hirschheim, 1996). Methods have a lot of components which specify: stages in a project, tasks to be carried out, outputs produced, constraints to apply, which people that could be involved, tools to be utilized and how to manage and control the project (Avison and Fitzgerald, 1995). Methods are devised to make the ISD process simpler and more controllable. For some years there has been awareness that methods do not solve all problems in information systems development. Huisman and Iivari (2002) comment that the widespread belief of adherence to information systems development methods is beneficial to the organization is still controversial. Recent surveys reports that many organizations claim that they either do not use any methods, or they used their in-house developed methods (Huisman and Iivari, 2002; Kiely and Fitzgerald, 2003).

Methods may have many different functions from the technical part of making the development process easier to control to the more political side where the methods may

*Hans Olav Omland, Agder University College, Service Box 422, N-4604 Kristiansand, Norway, hans.o.omland@hia.no

act as rituals (Robey and Markus, 1984). Methods may be understood and deployed differently depending on the context and the actual developer. Ørvik el al. (1999) describes four different understandings of the same systems development method related to an actual development. The first is the formal description of the method. The other three all occur in the actual deployment process, namely what the developer interpret, i.e. the understood method by the developer; how it is adopted in the organization, the adopted method; and how it is actually used, the method in use. According to Ørvik et al. (1999) these different understandings of the method will influence the use of it.

Information systems are not developed by methodologies, but by human actors, using their competence and methods in the development process. The concept of competence is used in many ways and in many different areas of research (Basselliar et al., 2001). Competence is the ability, the enabler, providing the means to performance (Basselliar et al., 2001; Ross, 2003; Fitzgerald et al., 2002). Basselliar et al. (2001) discuss different aspect of competence. According to them: "Regarding competence and performance as the same", leads to confusion. They state that the terms are related, but "factors other than competence – such as motivation, effort, and supporting conditions – may influence performance". Competence may also be considered as skills (Basselliar et al., 2001; Fitzgerald et al., 2002; Lee et al., 1995), personal traits (Basselliar et al., 2001), and as knowledge (Basselliar et al., 2001; Nielsen and Stage, 1994; Mathiassen and Purao, 2002). Marcolin et al. (2000) refers Kraiger et al. (1993) describing different conceptualization of an individual's competence, as cognitive competence, skills, and affective competence. Lee et al. (1995) defines four broad categories of critical IS knowledge/skills, technical, technology management, business functional, and interpersonal and management knowledge/skills.

Many authors mention communication competence, or skills, or abilities as one of the most if not the most important competence for a systems developer (Fitzgerald, 2002; White and Leifer 1986 quoted by Mathiassen and Purao, 2002; Fitzgerald et al., 2002, Cockburn, 2002). Other aspects of competence mentioned by Fitzgerald et al. (2002) are a kind of intelligence characterised by being compositional and intentional, rational, and having creative and analytical skills simultaneously. To be rational in this respect, i.e. for practitioners, is defined as being able to communicate in a way that the users understand what is being communicated as compared with being rational in relation to formal methods, as "doing the right thing in an efficient and logical way".

The concept of competence is one thing, use of it, performance, seems to be quite another. This may be analogue to the terms methods and methods-in-action (Fitzgerald et al., 2002), and espoused theories and theories-in-use (Argyris and Schon, 1978). These distinctions introduce the element of practice. Nielsen (1990) defines working practice as follows:

> "Working practice is the concrete actions that are actually taken in a particular situation in information systems development. Usually it is the working practice of the systems developers and others that take a professional part in the development efforts."

All development takes place in practice. Many things influence practice. Developers view of software development influence their actions (Cockburn 2002), i.e. how they solve problems and develop information systems. Cockburn (2002) also discusses extensively the problems of communication, particularly the activity of reconstructing information in a meaningful way (Cockburn, 2002). Communication may be hampered further by other contextual factors as domain knowledge, politics, personal ambitions, or communicating on different levels.

ISD is not just technical development, but the more social parts of development are typically not supported by traditional methods (Kiely and Fitzgerald 2003). Their data suggested further that in-house methods are preferred in developing information systems compared to external methods. Practitioners viewed external methods as big, rigid, and partly out-dated to fit the present more ad-hoc development environment.

Fitzgerald et al. (2002) suggest a framework for ISD Method Use. They discuss the different parts of the framework. Even if they name the relationships between the different components in their model, they say little about the relationships themselves.

This literature review shows that little is said about the relationships between developers' competence, methods, and practice in information systems development. This research intends to investigate more about these relationships especially in the analysis activities in the ISD process. Results from the analysis activities will influence the ISD process and the developed information system. The discussion above motivates the following research question:

"What do systems developers say about the relationship between developers' competence, methods, and practice in information systems development?"

The rest of this paper is structured as follows: the research method is described first; the case is described and analyzed. The themes from the analysis are then combined, followed by the conclusion.

3. RESEARCH METHOD

This research relies on a study of a specific information systems development project conducted by a business and IT consultant provider, here called BICT. The data are collected through semi-structured interviews (two hours each). The interviews were taped and transcribed. The five interviewees were central actors in the project and are working in the consultancy.

The research method reflects the intention of seeking knowledge of what happened in the relationships between developers' competence, method, and practice in the actual case. The analysis of the data was partly based on grounded theory (Glaser and Strauss, 1967). The grounded theory framework was used in the following way in this case study:

Open coding: This is the initial step where the data or text is opened and the different meanings of the interviews are sought. The interviews were transcribed and the text was coded. The research theme relates to developers' competence, methods, and practice. The coding reflected these wide categories, and was therefore not completely open.

Axial coding: Based on the coded text from the open coding, I searched for categories that can describe relations between the different codes and concepts. These were placed in main categories which, in combination constructs patterns or sets of axis explaining the data material.

Selective coding: The categories are combined into a coherent image of the data relevant to the chosen research theme.

My strategy has been open and emergent with the exception of the limitations I have mentioned in the open coding description. I heed the warning Walsham (1995) gives against ignoring any existing theory and let myself and this study be informed from relevant theory. The chosen research method relies heavily on the data which are the participants' interpretations of the situation and on my perception and interpretation of these data. This is the case in any research of social phenomena. Any generalization of the findings should therefore be handled with caution.

4. THE CASE - CONTEXT

The information system described is to be used to keep track of loans, down payments, lack of down-payment, changes in loans, loan guarantees, insurances payments and reimbursements for the same when down payments are made later than scheduled. Transactions are to be made both in Norwegian and foreign currencies. The information system contains four interrelated modules and was developed by BITC.

The development of the information system is based on a fixed price contract. The information systems requirements were developed in several steps. The process is described here as it had an influence on the development of the information system. BITC was involved in steps 2 – 5 that lasted for 3 months all together. The total requirements process was as follows:

1. Initial system requirements developed by a consultant company
2. BICT submitted an initial offer, based on the initial systems requirement
3. Analysis and development of the requirements by BICT and the customer
4. BICT submitted final offer based on system requirements made in step 3
5. Contract negotiations and signing of the fixed price contract

This final price was about 15 – 20 % higher that the initial offer. The negotiators from BITC considered this to be the maximum increase acceptable to the customer while some of the developers thought the increase did not reflect the calculated increase in cost for BICT.

During the above mentioned process the project was staffed with 6 developers, including three senior systems developers. After the contract was signed (point 5) the development of the system started. Two of the senior systems developers were then substituted with younger systems developers. The remaining senior systems developer became the project leader. He had a record of delivering projects on time and with good quality. After nine months he left the project and one of the initial senior systems developers re-entered the project group as project leader. Later the person responsible for the design of the solution left the project. This position was not filled as the project group considered his task to be fulfilled since a solution had been chosen before he left the project. During the busiest part of the design and programming faces round 15 people worked on the project. At the time of the interviews 5-6 people worked on the project. The project was originally estimated to take one year, but had, at the time of the interviews, lasted for two years with an overrun of 100% for man hours. The information system was not approved by the customer at the time of the interview.

5. THE CASE – ANALYSIS

The following categories emerged from the analysis of the data: Domain knowledge, analytical competence, prototyping competence, and validation competence. In the following I will describe these categories in detail.

5.1. Domain knowledge

The domain, where the information system is to be used, was perceived to be complex. The domain was complex in at least two ways, both how to solve the tasks as such and the use of terminology in the different departments at the customer's side.

> "There is a reason that a lawyer sits together with a political scientist, together with an economist, together they manage to cover the field."

The project leader tried to introduce an understanding of the basic process that took place in the company was more or less like a process in a bank:

> "It is that they make a number increase from one side and decrease from the other side"

He did not succeed in getting a common understanding with the users on the basic process that the information system should help them to control.

The information system had four modules serving four different departments. Some functions were common for all four departments and therefore included in all four modules. But the vocabulary to describe these functions was not necessarily the same across departments. It took time before the developers understood these differences. The lack of common use of terminology in different departments further complicated the development process leading to delays and more complex development process and information system. One developer commented:

> "Consequently, they continue with different views in this matter and get a complicated system, as opposed to that we got to simplify their view…we did not manage to change their view in this matter"

Better domain knowledge on the developers' side might have led to more professional discussions with the users. It took time before the developers understood what the users meant. Better understanding of the users at an earlier stage could have led to a better understanding of the information system and a clearer common understanding of the processes that took place in the company. One developer commented that the users were very good at solving their tasks, but he felt they lacked the deeper understanding of the processes related to the information system.

5.2. Analytical competence

The interviewees all knew that they should use RUP as development method in the project. The data do not state explicitly who decided that, but it must have been either the customer or BICT. The data report that RUP was not used as an overall method in the project. Wall graphs, use cases, UML, prototyping, and NIAM were used. Wall graphs were used for the initial analysis of the company and the processes to be covered by the information system. Many of these processes were described as use cases using UML notation. These use cases were used further into the analysis and design activities. Prototyping was used late in the analysis activities and in the design activities. NIAM was used mainly as a tool to check consistency in a developed part of the system.

All developers in the early phase of the project participated in the wall graphs sessions together with the users. The wall graphs were used to get to know the company and the routines used in the daily work. Use cases were intended to be developed from the wall graphs. To save cost most use cases were written by the users with some initial assistance from the systems developers. Since the use cases were written by the users some of the use cases turned out to be descriptions of existing routines. Developers reported that some use cases were very clear while others were very superficial.

> "The use-cases that were thin, there it was difficult to get hold of what was behind them; we did not understand it"

The use cases differed in quality from very general to very accurate descriptions with the more general descriptions mostly in the superficial use cases. The initial reaction at the developers' side was that the superficial use cases seemed easy to implement. They

discovered later that the use cases were not good enough for designing the information system. They had to do more analysis to get a good understanding of how the users worked. This proved to be a challenge as the developers had problems understanding the domain specific language of the users.

> "… you know that since you do not completely understand what she is saying; it is not only a question that I do not understand these things. But then there is probably something else I haven't understood, something else that I haven't got hold of……."

As time pressure increased an interface designer was employed to design interfaces. The idea was to shorten the development process by providing ready designed interfaces for the developers. The designer did not know the project well. When the developers started to implement his interface designs they discovered that the designs were not consistent and did not fulfil the systems requirements. They had to make their own analysis leading to change in the proposed design, and to more work for the developers.

> "I did not participate in writing that specific design document, because at that time we started to run out of time, so it was somebody else that should write the document when we did something else. And then the design documents fell on my lap as a starting point for develop it further."

Even though the developers had competence in writing use cases something happened in practice. The major reasons for this seem to be that the users did not give exact explanations of their work and that the developers did not know enough about the field to be able to ask questions.

> "We discovered that it was a little bit difficult to write the use case, because, especially the lady that knew about this said: 'I am doing many things'. They [the use cases] were very thin. So one gets a signal, yes, here we are more uncertain than."

Even if the developers had analytical skills/competence they failed in getting the information they needed for developing the information system or it took a long time to get hold of it.

5.3. Prototyping skills/competence

Prototyping is reported used in the wall-graph process, in making paper based interfaces, and in the interactive interface. Prototyping was used basically for getting information and understanding during the analysis activities, as in the wall-graph sessions and interface sections.

The data indicate that the developers had competence in prototyping, but that the prototype was developed too late in the process.

> "It is certain that it would have been useful for us to get the prototype earlier, it came too late in that phase"

The anticipated use of a prototype at an earlier stage is stated by one of the developers:

> "The prototype had certainly helped the users as well to see the relations between the things and that, in a way; we had got a common picture of how things really were connected"

The users became much more active in their feedback to the developers when they could press buttons on the interactive prototype. The data suggest that the prototype helped the developers in their understanding of the domain knowledge in the project creating a common understanding between the participants in the development. Since the results of using prototyping are reported as very positive, why didn't the developers use it earlier in the process? The data do answer this question. Developers reported that even

though prototyping was used, only one to two cycles were used in most cases. One reason for not continuing with more cycles was lack of time.

The interview data also include statements that prototyping could have been used in at least two additional places. The information system was designed with four distinct interdependent modules. These could have been prototyped in a way that would hinder errors moving from one module to the other. One developer reported that prototyping could have been used successfully in the module that produced letters. This module should produce many different letters. Instead of designing each letter separately the developer was occupied with finding similarities in the letters.

> "We shall use 90% of the time we are to use on producing letters, shall we use on producing the first letter. If we get a full understanding for producing letters, we will produce the rest in a very short time. I have not been able to implement this idea."

The data report that prototyping was very useful in the communication process, especially in understanding the users' needs in the interface design process.

5.4. Validation competence

The following were used in the validation process: prototyping, NIAM, questioning, the contract, use cases, the design document, systems test, and a method where the developers were asked to prove what they had made was wrong. Prototyping was used mostly in validating analysis information. NIAM was used to verify database design. Questioning was used in the analysis activities to verify the understanding of the information system by the developer. Control of the final result was done consecutively. The system test was done late in the process, together with the users. It turned out to be very difficult for the developers to prove that what they had made was wrong, so the project leader changed his method to get them to prove that what they had made would work.

The data suggest that the developers had the needed validation competence except for the proving their produce wrong.

Some of the validation work turned out to be difficult as the premises for validation were not well organized. The systems requirements were to be found in three different documents that were not coordinated: the contract, the use cases, and the design document. At times conflicts arouse between these documents leading to extra work to find out what requirements should be fulfilled. The contract was written in a way that left very little room for the developer to manoeuvre.

> "I think furthermore that the contract was very good for the customer. That they in a way could decide and change their mind all the time about what they wanted to have"

The contract also contained some statements that could be interpreted in different ways leading to unnecessarily complicated development and, according to the developers, did not improve the usability of the information system.

The information system test was conducted together with the users. The developers did not perform their own "internal" test first. The result of the user participation in the information system test was that "user-errors" were discovered, i.e. errors were of a more cosmetic type and basically related to the content of the interfaces like lead texts and layout.

> "… and they are occupied with completely different things than the functional. …. So you may not find functional errors or short comings before you are in the second or third circle.."

Tests were performed late in the development process. This way of testing had serious consequences: The developers did find the functional errors or short comings late in the testing process. It took time to correct the errors. Since some of the development was done in parallel errors discovered in one module of the information system had consequences for other modules, as code was re-used, i.e. errors had to be corrected several places leading to more work and further delays.

If the developers had produced one functional, error free prototype they could have multiplied it and made the adjustments needed to satisfy users' needs. It might have taken longer time to get the first prototype ready, but time and effort might have been spared in implementing other parts of the information system.

Some of the validation problems were due to lack in validation skills. Others seem to be caused by outside influences, like time pressure, poor planning, or the contract to mention a few.

6. COMBINING THE THEMES

Combining these four themes can improve our understanding of some underlying themes and to some extent improve our understanding of the relationships between developers' competence, methods and practice. In the following we combine these understanding and some earlier results with existing literature in the IS field. This is done in the following themes:

1. competence at two levels
2. life in the shadow of the contract
3. competence and methods

6.1. Competence at two levels

The data in section 5 show that the developers had competence, both concrete, technically, and on practice level. This is clearly seen in the analysis of prototyping activities. Even if they had competence in making a prototype in practice, it seems that they were not able to decide when to make and use the prototype. And this is really surprising as one should believe that making a prototype and knowing when to use it are two sides of the same matter. But there are, if we look at the data, a decoupling between these two points.

One could think that this is because there is a decoupling between competence and practice. And this may be a reasonable explanation. But the competence that is in prototyping is a competence that is put in practice. In this technique there is no decoupling between competence and practice. It is therefore not possible to explain this matter fully by stating that there is a bad relationship between competence and practice related to prototyping.

Another possible explanation is that there is no competence present concerning when to use prototyping. If this competence is not present, we can not talk about a decoupling between competence and practice. This may not be a full explanation because some kind of competence is always present like competence to discuss, at least, when something has to be done or when not to do it. But that competence may be of a more general character that does not specifically say when prototyping is useful or not in a given situation. So

having competence in what we can call the how to level in prototyping does not in itself lead to that we can talk about and have a reflective relationship to prototyping.

A third possible explanation is that related to prototyping techniques we have a very direct relationship between competence and prototyping method to practice. There is prototyping method and that is what has been applied here. Prototyping method is related to the how to level. Since the how to level also contains an element of method, the developers' competence, method and practice and the relationships between them are activated in this development. This may mean that the method they have used obviously had nothing to say about when and what should happen when applying prototyping.

The data show the same decoupling for other competences and methods like the wall graph section and the interface section. But also in these sections there is an underlying method that is very influential on practice.

This decoupling is present in the literature about information systems development with many references to techniques and methods, but at the same time rather few references about how one as a professional developer are reflecting on these methods and techniques. Mathiassen (1998) and Mathiassen and Purao (2002) describe "the reflecting developer" and stress that the most important for the developer is not the techniques and methods, but the ability and competence to reflect on the contribution these techniques and methods have in the development process and when they are put to use.

6.2. Life in the Shadow of the Contract

The data show that the possibilities to use competence, method and practice are very dependent on conditions in the given context. The contract was a fixed price contract with some specific regulations of how the users should cooperate with the developers in order to reduce the cost for the customer. Some of these regulations clearly reduced the relationship between developers' competence and practice, and sometimes even hindered the use of competence. This is seen in several situations.

First, after the wall graph sessions, where both users and developers participated, the users wrote the use cases with some initial help from the developers. The contract stated that the users should write the use cases and hindered the developers to utilize their competence in writing the use cases. The developers know the method and have competence in making use cases. This could not be utilized because the users should use a method they did not know, that they have had only some lecturing in using, and that they did not have any competence in. The data show that this had two consequences: a) many of the use cases that the users wrote were either copies of the old information system or very superficial, b) to late the developers discovered that some use cases were superficial.

The users did not have competence and method to improve the use cases and thereby take the opportunity to make the new information system better than the existing. The use case competence and method that the developers had were brought into practice late in the development process.

To late did the developers discover that the information system was much more complex than first anticipated. The discovery was made after they had started the design of the information system based on the superficial use cases. This situation can partly be explained by the contract that said the users should write the use cases. Another explanation may be that the developers did not have enough domain knowledge in the field and therefore did not enquire more into the superficial use cases. But this is not a

full explanation of the situation. A third possible explanation relates to the communication competence. Since the developers recognized that their domain knowledge did not cover the whole field, one should think that they would be more careful in their communication with the users making sure that they understood what the users said. The importance of communication competence in information systems development is supported by many writers (e.g. Mathiassen and Purao, 2002; Fitzgerald et al., 2002; Cockburn, 2002). A fourth possible reason may be that the users, during the development process, learned more and thereby was better to describe the information system later in the process than in the beginning.

Second, the contract stipulated a certain delivery date. To speed up the development process, the users and the developers established an eight-week schedule for the analysis, design, acceptance for the design and programming of each of the four modules. When put in practice it turned out that this schedule was too tight. The developers report that the users used very long time to give feedback to the design proposals. Part of the explanation is found in the contract giving the developers responsibility even for getting the feedback from the users. The contract thereby regulated practice and put all the responsibility on the developers also for things they had no control over. We see here that the contract had a direct influence on how this method was used in practice, leading to further delays. Two actions were taken to make up for the delay: a) hiring an interface designer from outside the project, and b) developing some of the modules in parallel.

The interface designer designed the interface while the developers worked on other parts of the project. The interface design was inconsistent and did not fulfil systems requirements. How can we explain this? It is natural to anticipate that the interface designer has competence and methods applicable in interface design. So a decoupling between competence, analysis and practice will not explain what happened. One explanation can be that systems requirements were difficult to find and understand as they were spread in three different documents, the contract, the use cases, and the design document. It seems reasonable to believe that the designer used the design documents in his design. As this did not include all requirements, and the designer was not familiar with the information system it was difficult for him to make a complete design even if the he brought his competence into practice. This does not explain fully what happened. Since the developers were busy doing other work the designer could therefore not get their views on the design and utilize their competence. This supports Curtis et al's (1988) findings quoting one system engineer: "Writing code isn't the problem, understanding the problem is the problem".

The decision to develop modules in parallel had some consequences that probably were not clear to the developers when the decision was taken. This seems strange. Some of the modules were depending on other modules. As modules were developed in parallel errors in some modules influenced other modules. It seems that the competence to develop the modules were present. How can we explain the practice that happened? It is surprising if the developers did not have competence to understand the dependencies between the modules and the consequences this would have in practice. So this does not seem to be an explanation. A possible explanation may be that the pressure on delivery led to a decision basically grounded in the time aspect leading to a decoupling of developers' competence and method from practice.

The same decoupling seems to have happened in the validation/testing activities where the systems test was performed together with the users, and systems requirements were spread over three documents.

6.3. Competence and Methods

BICT had earlier used an in-house method. Due to a merger and a decision to use RUP the company was in a transition period and a new method was not fully operational. RUP was to be used in this case. Most of developers had received little training in RUP, had little experience in using it, and did not feel competent to use RUP. RUP was therefore not used in practice in the project. Kiely and Fitzgerald's (2003) investigation into use of large and external methods shows that these methods are seldom used in information systems development in companies since they are perceived as big and complex by the developers. But the data here show that lack of use in practice also may be attributed to lack of competence. This may be explained in several ways.

Development is a group effort. There need to be both an agreement, a commitment and competence in the group to be able and willing to use a method in practice (Basselliar et al., 2001). The data show this these three elements were not present here, even if some of the developers had some knowledge in RUP, and the developers reported an understanding that they should use RUP. One possible explanation that they did not use RUP is that BICT did not educate the developers properly in the method. Another explanation may be the change of project leader in the middle of the project. It took some time before he got to know the project and therefore could give clear directions. Furthermore he did not know much about RUP. But this does not explain what happened fully since the project was already underway when he came into the project. A third explanation may be that the developers perceived the project as easier than then it turned out to be and thought they would be able to do it without using a big external method. When they discovered the complexity of the project they might have thought it to late to start RUP as that would have delayed the process even more. They ended up using no formal method in the project. Huisman and Iivari (2002) report that this is common in many organisations.

7. CONCLUSION

The major contribution of this study is the establishing of the relationships between developers' competence, methods and practice. Results confirm existing results in the field and offer new insights, especially related to the relationships mentioned above.

The study identifies competence at two levels, the how to level and the when and where level. This is clearly seen in the decoupling between the competence how to make a prototype and the competence to decide when and where to use it. The same decoupling is present in the wall graph and interface sections of the project.

Contextual conditions like contracts may influence and even hinder the use of developers' competence and methods in practice. These contextual conditions are made to serve some purposes, but do probably not consider the counter effects they have. These effects are documented here especially related to the contract and is also seen in the validation/testing activities.

The study confirms that lack of domain knowledge has great influence on relationships between developers' competence and practice. The study further confirms that lack of communication competence has similar influence on relationships between method and practice. This study identifies a project where both conditions were present.

The study identifies situations in development where decisions to rectify a situation hinders the use of developers' competence and methods in practice, often leading to counter effects. This is clearly seen in the attempts to speed up the project development.

The study confirms literature on little or no use of large and external methods when companies develop information systems. The reasons given for not using these methods are also confirmed.

REFERENCES

Avison, D. E., Fitzgerald, G., 1988, *Information Systems Development. Methodologies, Techniques and Tools*, Blackwell Scientific Publications.

Avison, D. E., Fitzgerald, G., 1995, *Information Systems Development: Methodologies, Techniques and Tools.* London, McGraw-Hill Book Company.

Argyris, C., & Schon, D., 1978, *Organisational Learning: A Theory of Action Perspective*. Addison Wesley, Reading, Mass

Bassellier, G. R., Blaize H., Benbasat, I., 2001, Information technology competence of business managers: A definition and research model, *Journal of Management Information Systems* **17**(4) pp. 159-182.

Cockburn, A., 2002, *Agile Software Development*, Addison-Wesley

Curtis, B., Krasner, H., and Iscoe, N., A field study of the software design process for large systems, *Communications of the ACM*, **31**(11), 1268-1287

Fitzgerald, B., Russo, N. L., and Stolterman, E., 2002, *Information Systems Development: Methods in Action*, McGraw-Hill Companies

Glaser, G.B., and Strauss, A.L., 1967, *The discovery of Grounded Theory – Strategies for Qualitative Research*, Aldine De Gruyter, New York

Huisman, M., and Iivari, J., 2002, The individual deployment of systems development methodologies. *CAiSE* 2002, Springer-Verlag Berlin.

Iivari, J., and Hirschheim, R., 1996, Analyzing information systems development: A comparison and analysis of eight development approaches, *Information Systems Journal* **21**(7) pp. 551-575.

Kiely, G., and Fitzgerald, B., 2003, An investigation of the use of methods within information systems development projects, *In Proceedings of IFIP WG 8.2 Conference*, Athens Greece

Kraiger, K., Ford, K., and Salas, E., 1993, Application of cognitive, skill-based and affective theories of learning outcomes to new methods of training evaluation, *Journal of Applied Psychology* **78**(2) pp. 311-328.

Lee, D. M. S., Trauth, E. M., and Farwell, D., 1995, Critical skills and knowledge requirements of IS professionals: A joint academic/industry investigation, *MIS Quarterly* September 1995 pp. 313 - 340.

Mathiassen, L., 1998, Reflective systems development, *Scandinavian Journal of Information Systems* , 10, (1-2)

Mathiassen, L., and Purao, S., 2002, Educating reflective systems developers, *Information Systems Journal* **12** pp. 81-102.

Marcolin, B. L., Compeau, D, R., Munro, M. C., and Hupp, S. L., 2000, Assessing user competence: Conceptualization and measurement, *Information Systems Research* **11**(1) pp 37-60.

Nielsen, P., 1990, *Learning and Using Methodologies in Information Systems Analysis and Design*, Ph D thesis Lancaster University

Nielsen, P. A., and Stage, J., 1994, The software designer: Skills and change. *IRIS*, University of Oulu, Finland, Department of Information Processing Science, University of Oulu, Finland.

Robey, D., and Markus, M. L., 1984, Rituals in information system design, *MIS Quarterly* (MARCH 1984) pp 5-15.

Ross, J. W., 2003, Creating a strategic IT architecture competency: Learning in stages, *MIS Quarterly Executive* **2**(1) pp. 31 - 43.

Walsham, G., 1995, Interpretive case studies in IS research: nature and method, *European Journal of Information Systems*, (4:2) pp. 74-81

White, K. B., and Leifer, R., 1986, Information systems development success. Perspectives from project team participants, *MIS Quarterly* **10**(3) pp. 214-223.

Ørvik, T. U., Olsen, Dag H., and Sein, M. K., 1999, *Deployment of System Development Methods. Exploring Paradigmatic Mismatches. Evolution and Challenges in System Development*, Zupancic e. al., Kluwer Academic/Plenum Publishers pp. 19-31, New York.

The relationships between competence, methods, and practice in information systems development

Hans Olav Omland

University of Agder, Kristiansand Norway
Hans.O.Omland@uia.no

Abstract. This paper investigates the relationships that unfold between an actor's competence, methods, and practice during information systems development (ISD). The data was gathered in a case study of a successful ISD in a Norwegian municipality. In theory, competence, methods, and practice are **separate** and clearly distinct elements. In actual ISD, however, the three elements form close and integrated relationships. While previous research has addressed some of the relationships between competence, methods, and practice, researchers have yet to describe fully how the three elements relate to and influence each other. This paper's main contribution is a new and more detailed understanding of the tight and intrinsic relationships between competence, methods, and practice and how the three elements dynamically influence each other during ISD processes. The result is a deeper understanding of the ISD process that will help systems developers better establish, monitor, and succeed in their ISD projects.

Key words: Competence, methods, practice, systems development.

1 Introduction

ISD is "an intentional change process which is driven by certain more or less clear objectives" (Mathiassen 1998 p. 70). Mathiassen (1998) further states that while actors perform the change process in a context that includes a set of social and technical factors, the change process itself is shaped and influenced by many factors, "including the experiences and competence of the development group" and "the dynamics of the objectives" (p. 70). Researchers have made many attempts to describe ISD methods. Fitzgerald et al. (2002) coined the "Method-in-Action" concept to denote how a method "is uniquely *enacted* by the developer" (p. 13). This enactment, according to Fitzgerald et al. (2002), is shaped by the development context and influenced by the role of methods. Madsen et al. (2006) extend this line of research, suggesting a framework that explains how a unique and local method emerges over time in a complex interplay between human action, structural elements, and the "emergent method".

Although both Fitzgerald et al. (2002) and Madsen et al. (2006) clearly centre their research around method, the scope of their ISD discussions widen to include both the developer and other contingency factors. This widening of scope recognises that, while systems development might be informed by methods, it depends on much more than methods.

Human actors develop information systems using whatever competence they have in the chosen methods or method elements. Competence and methods, used in practice, are key elements in ISD processes. It is therefore important to understand and describe the relationships between them in actual ISD situations. Some research describes the relationship between methods and practice (e.g. Fitzgerald et al. 2002; Madsen et al. 2006) or competence and methods (e.g. Mathiassen and Purao 2002); there is less existing work on the relationship between competence and practice (e.g. Mathiassen and Purao 2002). Section 2 further reviews the literature on these relationships. No research has been found that describes all three relationships and how the elements influence each other. This research therefore seeks to answer the following question: How do competence, methods, and practice relate to and influence each other in ISD?

This paper reports on a case study of a successful ISD for a Norwegian municipality. The case data and analysis of it form the basis for describing the relationships between these three elements in the ISD project.

The paper is structured as follows: The next section presents an overview of research on competence, methods, and practice and the relationships between them. Section 3 describes the research approach. A case description and case analysis follow in sections 4 and 5, respectively. The relationships between competence, methods, and practice are then discussed in section 6, followed by the conclusion in section 7.

2 Methods, Competence, and Practice

2.1 The Elements

Methods developers typically devise methods to make the ISD process simpler and more controllable. Fitzgerald et al. (2002) define a method as "*A coherent and systematic* approach, *based on a particular* philosophy *of systems development, which will guide developers on* what *steps to take,* how *these steps should be performed and* why *these steps are important in the development of an information system'"* (p. 5, italics by authors). In this paper, methods are understood to cover larger or smaller parts of ISD. Also, the term "method elements" is sometimes used to describe parts of a method.

Methods differ dramatically and often address different objectives (Avison and Fitzgerald 1995). They are also based on many implicit and explicit assumptions and views (Iivari and Hirschheim 1996). Iivari and Lyytinen (1998) analyse 10 Scandinavian ISD approaches using concepts such as scope, value orientation, knowledge interest of ISD, the role of methods, and the principle of the ISD process (p. 162). Because the focus of this paper is to research the relationships between methods, competence, and practice in ISD, it is important to investigate how developers view and use methods. Ørvik et al. (1999) describe four versions of the same method depending on how it is understood and deployed. The first version formally describes the method, while the other versions relate more to its actual deployment - that is, how the developer interprets and understands the method, how the organization as a whole adopts it, and how it is actually enacted in an ISD process. Smolander and Rossi (2008) suggest that ISD actors might benefit more from "tools that help to identify and process the emerging conflicts than tools that aid in developing a technically 'perfect' and optimized solution" (p. 37).

Competence is deployed by human actors in ISD processes. The competence concept is used in many ways and in different areas of research (Bassellier et al. 2001), and many different conceptualisations are suggested. Still, researchers seem to agree on a generic conceptualization of an individual's competence as a combination of three elements: cognitive competence, skills, and affective competence (Marcolin et al. 2000 referring Kraiger et al. 1993). These three categories entail three important abilities: cognition (the ability to think); skills (the ability to do something); and affections (the ability to relate to other people). Lee et al. (1995) define four broad categories of critical skills and knowledge requirements for IS professionals: a) technical, b) technology management, c) business functional, and d) interpersonal and management knowledge/skills. Categories a - c relate to both cognitive competence and skills, while d) relates to affective competence. White and Leifer (1986) suggest five competencies that contribute to successful systems development: business knowledge, good communication skills, technical expertise, analytical skills,

and good organizational skills. These competencies group along the generic conceptualisation of Kraiger et al. (1993).

Competence is the ability, or enabler, that provides the means for performance (Bassellier et al. 2001). According to Bassellier et al. (2001), using competence and performance interchangeably will lead to confusion. Referring to Schaumbach (1994), they state that the terms are related, but that "factors other than competence - such as motivation, effort, and supporting conditions - may influence performance" (p. 162). This research adopts the notion of competence as the "ability to" think/analyze, do something, and relate to other people. These abilities belong only to individuals; an organization's abilities to perform depend on the individuals present in an organization at any given time.

Communication competence is regarded as important, and might be the most important competence in ISD (Cockburn 2001; Fitzgerald et al. 2002; White and Leifer 1986). Still, there is no agreement on what communication competence includes. Fitzgerald et al.'s (2002) discussion on being "rational" in ISD might serve as an example. A "rational" developer acts "in a way that clients and users understand" (p. 126) while being rational in relation to formalized methods is often referred to as "doing the right thing in an efficient and logical way" (p. 125) . Communication between developers and users will also involve domain competence. Both Truex et al. (2000) and Walz et al. (1993) stress the importance of developers having domain competence to ease communication with users in ISD processes.

Competence and experience are related concepts. Experience might lead to competence, but this is not an automatic progression; as described in *Reflective Systems Development* (Mathiassen 1998), reflection might be needed as well.

Practice is often referred to as something distinct from both methods and competence. ISD often occurs in situations that are complex, uncertain, unstable, and unique (Mathiassen 1998). As Mathiassen notes, these situations are often laden with value-conflicts, in which individual actors and different categories of actors participate in the important and difficult work of creating a common understanding of both the task at hand and how to reach the stated goals. According to Fitzgerald et al. (2002), the developer uniquely enacts the Method-in-Action. In Mathiassen and Purao's (2002) view, "… it is more important to have specialized knowledge about problems and possible solutions than it is to have general knowledge on how to structure and conduct development processes" (p. 83). This paper maintains that practice is what actually happens in development, rather than what ought to or should happen according to the method or the competence deployed.

2.2 The Relationships

The following focuses on the relationships between methods and practice, methods and competence, and competence and practice.

Methods/Practice. Because formalised methods are devised to inform ISD practice, it is reasonable to expect that methods are widely used and that they

contain advice on how actors should implement them in practice. However, method designers offer little practical advice on implementing methods in practice (Fitzgerald et al. 2002). And many organizations claim that they either do not use any formalized methods or that they use methods developed in-house (Huisman and Iivari 2002; Kiely and Fitzgerald 2003). This is surprising as it is widely believed that system developers' adherence to methods in ISD will benefit the organization (Huisman and Iivari 2002). On the contrary, Ciborra (1998) suggests that even if method is one of the ISD discipline's key features, it is also probably the "true origin of its crisis" (p. 8). Wastell (1996) suggests that methodology can act as a social defence, undermine the learning process, and hinder creativity in ISD processes. Truex et al. (2000) question whether ISD methods really describe what happens in ISD practices. Their view is that if actors view practice through a method's concepts, things that happen only in practice are not noticed or registered unless they are formal concepts in the methodical arsenal. They therefore question the privileged view that ISD "is a managed, controlled process" (p. 60). Ciborra (2002) furthers this thought, introducing the term *Bricolage* to describe what happens in an ISD process; actors creatively use whatever is at hand during development. Walz et al. (1993) observed an ISD team and were surprised by how difficult it was to communicate and to achieve a common understanding of the team's tasks during a requirements determination process. Curtis et al. (1988) reference Zalkowitz et al. (1984), stating that there are "discrepancies between the state of the art and the state of practice in using software engineering tools and methods" (Curtis et al. 1988, p. 1268). Kautz et al. (2007) discuss persistent problems and practices in ISD. Although their focus is not specifically on the relationship between method and practice, it is obvious from their discussion that this relationship remains a complicated one in ISD. Methods are still promoted as solutions to the ISD problems, but their deployment don't necessarily lead to successful systems. This might be because formalized methods seek to avoid relying on individual developers' abilities in ISD processes (Fitzgerald et al. 2002). In their Method-in-Action framework, Fitzgerald et al. (2002) discuss the ISD components in detail, while simply suggesting the components' relationships and influences.

Madsen et al. (2006) study the emergent method, which they define as "*the actual unfolding development process and the activities, and applied method elements that comprise the process*" (p. 226, italics by authors). Madsen et al. (2006) see the development process as a sequence of activities and argue that their emergent method goes beyond Fitzgerald et al.'s (2002) concept of Method-in-Action "as it places more emphasis on what actually happens over time than on the relationship between the prescribed and the actual" (Madsen et al. p. 226). They consider the actual development process a result of "a complex web and interplay of enacting and interacting actors and structures" (p. 226). Their analytical framework draws on three perspectives:

- The *structuralist* perspective relates to the structural characteristics of systems development concepts.

- The *individualist* perspective reflects how the individual developer influences and shapes the emergent method.
- The *interactive process* perspective counts for the method's dynamic emergence over time.

The result is "the emergent method and information system under development" (Madsen et al. 2006, p. 228). Madsen et al. (2006) assume a more holistic view of systems development than Fitzgerald et al. (2002), but they still concentrate their description and discussion more on the emergent method and its use in practice, and less on the relationships in play during ISD.

Method/Competence. The relationship between method and competence has received considerable research interest. Some researchers state that methods are formalized competence. The advantage of this view is that competence is not needed to implement the actual methods in practice. Others state that developing information systems is both a technique and an art, or a creative process (Brooks 1987). Fitzgerald et al. (2002) further this argument, stating that if the developer follows the method strictly, it might preclude innovation. They go on to assert that, because ISD is a creative process, it is important that the individual developers engage their competencies (Fitzgerald et al. 2002). Finally, Fitzgerald et al. (2002) state that developers learn by engaging in methods, but they do not discuss or incorporate this perspective in their "Framework for ISD Method Use".

Jayaratna (1994) points out that a possible clash between the Weltanschauungs of the method creator and method user will lead to the latter using the method in a way that differs from the creator's intentions. The use of methods will be influenced by both developers' competence and their views of software development (Cockburn 2001). Ørvik et al.'s (1999) "the understood method" can be achieved only by some kind of relationship between method and competence. Necco et al. (1987) comment on this relationship, stating that ISD's key factors are improved involvement and better personnel; method in itself does not suffice.

Competence/Practice. Even though this relationship generates few hits in literature searches, the research literature directly or indirectly recognizes its existence. Madsen et al. (2006) stress "the importance of understanding the context, ….., the developers' preconceptions and actions and their interactions with other stakeholders, as well as the influence that these concepts have on the ISD process" (p. 227). To achieve this "understanding" of what actually takes place in ISD processes, competence must relate to practice. But Madsen et al. (2006) do not discuss how actors use this understanding in an actual development situation. In studying how systems developers work in practice, Westrup (1995) suggests that their representations of organizations are actively constructed as rational, coherent, and fitting to computerization. Developers use their competence to analyse and form their understanding of the actual situation in practice. The reflective practitioner uses competence to reflect on practice, contemplating both how to proceed in practice and what learning might occur as a result of practical experiences. (Mathiassen 1998).

Table 1 presents a summary of the research literature. The table is constructed to show the bi-directional relationships between method, practice, and competence. Descriptions of the relationship between method and competence were not found in research literature.

	Relationships		
	Methods	*Practice*	*Competence*
Methods	--	Fitzgerald et al. (2002) Huisman and Iivari (2002) Kiely and Fitzgerald (2003) Mathiassen and Purao (2002)	
Practice	Fitzgerald et al. (2002)	--	Fitzgerald et al. (2002) Mathiassen and Purao (2002)
Compe-tence	Brooks (1987) Fitzgerald et al. (2002) Jayaratna (1994) Madsen et al. (2006) Mathiassen and Purao (2002) Necco et al. (1987)	Fitzgerald et al. (2002) Madsen et al. (2006) Mathiassen and Purao (2002) Westrup (1995)	--

Table 1: A summary of the reviewed literature.

3 Research Approach

3.1 Research method

Since the focus of this research is exploratory and descriptive, a case study approach is selected. The case study investigates "a contemporary phenomenon within its real-life context" (Yin 1994, p. 13) where "the investigator has little control over events" (Yin 1994, p. 1) and therefore cannot manipulate relevant behaviours. The research answers a "how" question - in this case, How do competence, methods, and practice relate to and influence each other in ISD? This is in line with Yin's (1994, p. 6) criteria for a case study research strategy. Further, the case study approach gives the actors involved opportunities to describe their own and other actors' competence, methods, and practice in rich terms.

The unit of analysis is the organizational level. During the data analysis, it became clear that the research must also include the individual level to

adequately understand and describe the relationships between competence, methods, and practice.

Hereafter, the developer organization is referred to as DeveloperOrg, while the user organization is called UserOrg. Data was collected through document study and semi-structured interviews, which were conducted in retrospect after the project's main part was implemented. There were six interviewees from DeveloperOrg: the project manager; the product managers responsible for the ERP system, the invoicing system, and the e-procurement system, respectively; and two domain experts engaged in the project. There were nine interviewees from UserOrg, including the project manager, the project coordinator, and the subproject managers. The interviews were tape recorded and later transcribed, and the transcriptions were sent to the interviewees for validation. The researcher received feedback on the transcribed interviews via e-mail. All of the email comments related to minor issues in the transcription.

The data analysis used grounded theory techniques (Glaser and Strauss 1967) as follows.

Open coding: The transcribed text was coded based on the competence, methods, and practice of the seed categories' actors, and was therefore not fully open.

Axial coding: The relationships between the elements were coded using the open coding's coded text. Three different relationships emerged: competence/methods, competence/practice, and methods/practice. In axial coding, each element's influence on another element was also coded. This led to six different directions of influences between methods, competence, and practice.

Given the initial findings from the document study and interview data, two coherent reports were created describing the UserOrg and DeveloperOrg development stories, respectively. The reports were sent to the interviewees at the relevant organizations to validate the initial findings. An interview was then conducted with UserOrg's project manager to get feedback on the UserOrg report. Feedback on the reports was also received via e-mail from both UserOrg and DeveloperOrg. Again, the comments related to minor issues.

Selective coding: After receiving feedback from UserOrg and DeveloperOrg, additional coding was done and categories were combined into three topics that explained the relationships and influences between competence, methods, and practice: "Intrinsic Dynamic Relationships", "Common Understanding", and "Organizing Vision".

3.2 Case Background

UserOrg, a large (by Norwegian standards) local municipality, needed to replace its existing ERP system because the system vendor had announced that it would discontinue product support. UserOrg was searching for an ERP system that integrated accounting, budget, salary and personnel, invoicing, invoicing module feeding systems, and an e-procurement module.

UserOrg's IT manager organized the project internally; it included a steering committee with high-level officials to get easy access to decisions on financial matters, and project subgroups for each system module. Later, when the project entered the actual development phase, an informal project group was formed consisting of the IT project manager, a project coordinator, and all subproject managers. The externally hired project coordinator assisted the project manager, participated in project group meetings, and modelled work processes. The project manager and the project coordinator synthesized the different subproject groups' requirements specifications into one common tender document. Table 2 offers an overview of the project's main activities:

Time	*Activity*	*Comments*
Sept./Oct. 2001	UserOrg started internal process	Started as a substitution project
Feb. 21, 2002	Approved tender document	Developed process-oriented requirements specification
Feb. 28, 2002	Pre-qualification ended	Qualified two Developerorgs
March 2002	Demo-days	Two Developerorgs and UserOrg participated
April 10, 2002	Deadline for preliminary bid	Received two bids
May–Oct. 2002	Clarification of bids	UserOrg clarified bids with each of the Developerorgs
Oct. 2002	Final and best bid submitted	Reviewed by UserOrg
Feb. 2003	Contract signed	
May 2003	Development and implementation project started	Very close, active contact between UserOrg and DeveloperOrg
Jan. 1, 2004	All major ERP systems in production	Successfully implemented and set in production
Jan.–Sept. 2004	E-procurement and invoicing systems in development	Many new ideas and improvements
End 2004	E-procurement and invoicing systems in production	Successfully implemented and set in production
Spring 2005	All system modules in production	Systems development regarded as a success

Table 2: Development timeline.

After the bidding process was concluded, the winning DeveloperOrg organized a project group consisting of a project manager, an ERP manager, an e-commerce manager, and implementation-process consultants. DeveloperOrg considered UserOrg a very important user of its system. Because UserOrg had very high domain competence in parts of the invoicing

system domain and was to become the largest local municipality to install and use DeveloperOrg's entire ERP system package, DeveloperOrg took the opportunity to upgrade and improve its ERP system. The results of this project had positive effects on DeveloperOrg's market position in Norway.

4 Case Description

The following case description reflects the three seed categories of competence, methods, and practice.

4.1 Competence

In analyzing the interview data, five competence categories were identified: domain competence, project competence, IS development competence, negotiation competence, and communication competence. Table 3 shows the similarities and differences between UserOrg and DeveloperOrg.

Competence	*UserOrg*	*DeveloperOrg*
Domain	Most actors had worked for many years in their specific domains	Actors had developed IS for local municipalities for many years
Project	Four of five central actors had previously participated in a large ISD project	Actors regularly worked on large ISD projects
IS Development	Little experience	Several actors had considerable experience and education in the field
Negotiation	Competence at management level and sought advice from a buying specialist	Competence at different levels, including the ERP-responsible
Communication	Good	Good

Table 3: Project competencies.

Domain Competence. One UserOrg actor was a leading domain expert in the invoicing system domain, and was specifically sought out by DeveloperOrg. Both UserOrg and DeveloperOrg were expecting high domain competence from each other and both report that their expectations were met. This shared domain competence seemed to make communication easier within the domains.

Project Competence. UserOrg had changed ERP systems in the mid '90s. UserOrg's central actors and their project manager - who also managed the previous ERP project - had reflected on the earlier project's experiences and used their project competence to design this ISD project's main activities.

Development Competence. UserOrg and DeveloperOrg had different development competencies. This led to different interpretations of certain

incidents. The differences were especially visible in how they communicated in critical situations during the ISD.

Negotiation Competence. UserOrg and DeveloperOrg were continually negotiating requirement specifications. Negotiation competence was therefore an important competence in the project, and was seemingly balanced between the two organizations.

Communication Competence. Overall, both UserOrg and DeveloperOrg displayed high communication competence. In several incidents, however, actors in the two organizations failed to clearly communicate and this led to misunderstandings. This was especially visible during prototyping.

4.2 Methods

Neither organization used formalized ISD methods. Still, as Table 4 shows, their project efforts included several method element categories: brainstorming, tender document development, demo-days, requirement and contract process, and ISD processes.

Methods/ method elements	*UserOrg tasks*	*DeveloperOrg tasks*
Brainstorming	Elicited initial requirements specification	
Tender Document Development	Created a common tender document to help select winning bid	Developed and submitted bid document
Demo-days	Clarified functions availability and used demonstrations to help select winning bid	Presented their solution and fielded questions from UserOrg actors
Requirement and Contract Process	a) Clarified bid documents and requirements for new system b) Negotiated terms and signed contract	a) Clarified bid document and requirements for new system b) Negotiated terms and signed contract
Development Processes	Dynamically elicited requirements specifications by: a) Performing and discussing daily tasks b) Testing prototype, making suggestions, and giving feedback c) Acting as a pilot user	Dynamically elicited requirements specifications by: a) Observing and discussing the UserOrg tasks with UserOrg b) Developing and testing the prototype c) Observing the pilot user d) Acting as middle-man

Table 4: Method elements used in the ISD project.

Brainstorming. The project's initial activity was to identify what the new system should do for the different departments at UserOrg.

Tender document development. The project coordinator modelled and documented the different departments' requirements specifications and - through "a process-oriented tendering process" - merged those requirements specifications into a complete tender document for the whole system.

Demo-days. Two pre-qualified development organizations presented solutions to a case that UserOrg designed. UserOrg's different subproject groups participated in the presentations related to "their" system modules.

Requirements and contract process. After the demo-days demonstrations, each subproject group separately continued discussions with the competing development organizations to clarify what was ready for delivery, what was in the pipeline, and what the development organizations were willing to develop to satisfy UserOrg's requirements. This activity produced the system's initial requirements specifications.

After selecting the winning bid, UserOrg requested that their requirements became part of DeveloperOrg's standard system. However, DeveloperOrg had to be careful not to introduce changes that would adversely affect their existing customers' system usage. It therefore handled the UserOrg request as follows: If DeveloperOrg developers found a proposed requirement beneficial, they would integrate it into the existing system. If proposed requirements did not fit into established plans, the developers first tried to find ways to fulfil the requirements directly within the existing system. If that proved impossible, they would look for a way to work around the requirement within the existing system. Requirements that remained unmet after these two steps were put on a prioritized list - according to usefulness and importance for DeveloperOrg - that was used in requirements negotiations.

The company that ultimately won the contract had decided early in the process that it would win and would make the delivery a success. The project was anchored in DeveloperOrg's top management; indeed, representatives of top management were members of the project group tasked with preparing the final bid.

> "But then, actually, then it had such high priority or focus with us that the final bid was prepared by the Managing Director. And I may say the working chairman of the board and me and the salesman at that time and another person." (ERP-responsible, DeveloperOrg)

Development processes. The initial requirements specification was a starting point for a further dynamic specification elicitation that occurred through close interaction between DeveloperOrg's domain specialist consultants and UserOrg's users. Based on their common suggestions, DeveloperOrg's module consultant sent suggestions in writing to DeveloperOrg's module-responsible. She then decided what to include in the requirements specifications and instructed the programmers accordingly.

In addition to this more "formalized" method element, the organizations used several informal method elements, including:

- DeveloperOrg's domain specialist, who was also the system-responsible, communicated directly with an actual UserOrg user on one side, and with DeveloperOrg's e-procurement system programmer on the other.
- DeveloperOrg's invoicing systems consultant (who wasn't a domain specialist) communicated UserOrg's requests and ideas to DeveloperOrg's ERP-responsible.
- The ERP-responsible communicated directly with UserOrg's representative.
- A DeveloperOrg domain specialist communicated directly with both a UserOrg consultant and DeveloperOrg developers/programmers (in cases of emergency). As the support-responsible from DeveloperOrg put it, much depends on the size of the problem:

"If it is the calculations that fail completely, and 5,000 bills are to be issued tomorrow, we have to 'turn on the dime' and then just jump all formalities … try to get in the back door and solve the problem and get a new application to the UserOrg as soon as possible. So you are in the informal organization."

As the above examples show, even when there were agreed-upon methods for communications between UserOrg and DeveloperOrg, the methods weren't always followed.

4.3 Practice

Table 5 describes two major activities performed in UserOrg and in DeveloperOrg.

Activities	*UserOrg*	*DeveloperOrg*
Require-ments elicita-tion	a) Produced demo-case and participated in demo-days b) Negotiated and employed contractual legal expertise c) ERP system development d) Pilot installation	a) Demonstrated their system based on demo-case b) Negotiated and strategically handled requirements specifications c) ERP system development d) Pilot installation
Staffing	a) Deployed domain competence b) A "non-domain" actor cooperated with DeveloperOrg to obtain domain competence	a) Deployed domain and development competence b) Gained domain competence through cooperation with UserOrg c) Chose a "non-domain" actor based on relationship to UserOrg

Table 5: ISD activities.

Different actors had different perspectives on the usefulness of demo-day presentations. According to the UserOrg project manager, having demo-days was a "conscious decision" with the purpose of exposing weaknesses in the system and determining what type of solution the two developer organizations could deliver.

UserOrg's project coordinator argued that it was important to balance power between the UserOrg and the DeveloperOrg in the ISD process; demo-days could help achieve this by making DeveloperOrg present solutions to problems UserOrg wanted solved. "Using a demo case gives UserOrg the lead," he said.

As the following quotes from demo-days participants show, not everyone viewed the activity as beneficial.

> "… both DeveloperOrgs had too little time to prepare (for the demonstration of the case).... At the time of the demo, it did not benefit us much. It didn't." (subproject manager, UserOrg)

> "Use the exact data provided by them (the UserOrg) and try to reproduce the situations and demonstrate the processes they are looking for. As usual, you get too short a time. I remember that we did not get through it all." (representative, DeveloperOrg)

> "… it is often difficult to tell about the good news if you have to follow a big demo case from A to Z.… Such a demo may be very fragmented, making it difficult for the one who decides on what system to choose." (representative, DeveloperOrg)

Furthermore, information about the demo-days' purpose was presented only to UserOrg actors, not to those at DeveloperOrg. Still, as the interview data clearly shows, the demo-days' goals were neither understood by UserOrg's actors nor were they achieved.

When the requirements elicitation process began, both parties shared an interest in eliciting the best requirements. Later, DeveloperOrg used the requirements strategically during negotiations to win the contract and during the ISD processes.

UserOrg and DeveloperOrg deployed development and domain competencies in the ERP system development and the invoicing system's pilot installation. In the ERP system development, a high level of shared domain competence between DeveloperOrg and UserOrg made communication easy between the actors. DeveloperOrg basically handled the development technicalities, and the differences between DeveloperOrg and UserOrg in development competence (high and low, respectively) did not negatively affect UserOrg on this part of the ISD.

DeveloperOrg uses prototyping for major module revisions and to develop new modules, including (in this case) the invoicing system domain. UserOrg initially had high domain competence and low development competence, but the domain specialist went on sick leave shortly after the project started. UserOrg's substitute had low competence in both the development and the

invoicing domain. When the invoicing system development started, DeveloperOrg had no domain specialists available. Staffing of the invoicing system ISD group was therefore partly based on the good relationship between some UserOrg actors and a DeveloperOrg consultant who had high development competence and low invoicing domain competence. In that situation, relationships were more important than domain competence. A main actor from each organization cooperated in the development and improved their domain competence from low to high in the invoicing domain by developing the module together.

The differences in development competence turned out to be a challenge in the prototyping situation. However, DeveloperOrg was used to problems with prototyping and pilot installations in development projects:

> "We had some technical problems during the project. But we have that. We anticipate that we always have (problems) in projects…. we do not experience that as something critical". (Consultant, DeveloperOrg)

Reports from UserOrg contrast with this view. UserOrg actors generally felt that there were too many errors in the system during prototyping development. According to the DeveloperOrg representative, however, UserOrg's actors may not have understood the pilot user role or its implications. While the representative said that "it was entirely natural" that UserOrg members should be pilot users, it seems that "they were not conscious that they were pilot" on the invoicing system modules.

Indeed, when asked, one of the UserOrg actors said that he did not know that he was a pilot user. Differences in development competence between DeveloperOrg and UserOrg seem to lead to differences in interpreting the actual situation and the related activities.

5 Case Analysis

5.1 Relationships

The three categories - competence, methods, and practice - not only relate to each other, but also influence each other. The following discusses and exemplifies the influences between categories for DeveloperOrg and UserOrg. The bi-directional relationships are illustrated in Figure 1.

Competence influences Methods. An important DeveloperOrg objective was to win the contract for delivering the new system. As described (section 4), DeveloperOrg brought development and negotiation competence into play and used requirements specifications strategically as a method to conclude the contract negotiations.

UserOrg's project leader and most of the subproject leaders had previously acquired project competence in a large ISD project. This was clearly visible in how they chose and carried out brainstorming, tender document development activities, and the demo days.

Methods influence Competence. UserOrg domain specialists often put considerable energy and time into creating their own requirements specifications prior to acquiring new systems. Given this, DeveloperOrg consciously used requirements specification elicitation as a method for getting good ideas about how to create functions or improve existing functions in their systems.

UserOrg did not have competence in prototyping as an ISD method. However, as they engaged in prototyping, their competence in both the domain area and in the ISD method increased.

Competence influences Practice. DeveloperOrg's domain and technical competence let them tailor their bid to UserOrg's requirements specifications and thereby fulfil UserOrg's wishes within the project's technical and financial constraints. While doing this, they made sure that system changes had little or no adverse affect on the existing system users' daily and future practices.

Individual UserOrg users experienced increased system competence as the ISD led to some changes in how they used the new system.

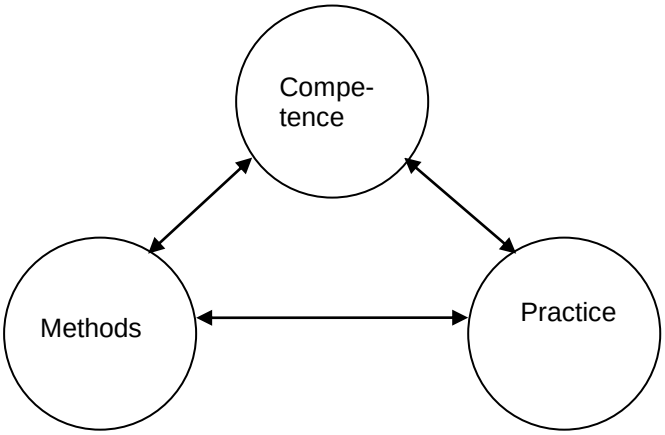

Figure 1: Bi-directional relationships between competence, methods and practice.

Practice influences Competence. Although it did not perceive a need for them, DeveloperOrg developed and installed special functions requested by UserOrg. UserOrg did not give any feedback to DeveloperOrg about these functions. Reflecting on the lack of feedback, DeveloperOrg's competence in requirements elicitation increased: Its developers will better scrutinize requirements elicitation and proactively seek out feedback from future client organizations.

Early in the ISD, UserOrg either received no installation manuals, or the manuals they did receive were insufficient for system installation. Once they requested and received better installation manuals, they made fewer mistakes and increased their competence in later installations.

Methods influence Practice. In developing the e-procurement system, DeveloperOrg's representative consciously chose to use observation and discussion as a method (see Table 4's "Development Processes" section). This influenced how UserOrg and DeveloperOrg representatives worked together in practice. DeveloperOrg's representative reports that he was surprised by how UserOrg's e-procurement-responsible used some functions very differently than how the system designer intended.

A UserOrg representative participating in the e-procurement activities said that the choice and use of the development method influenced her work in practice.

Practice influences Methods. In emergency situations, DeveloperOrg's support-responsible took shortcuts, using every possible way to fix a problem. In doing so, he disregarded the predefined methods for correcting system malfunctions. The support-responsible was thereby able to solve emergency problems faster than if he had reported the error using the prescribed method.

Because some UserOrg representatives found the demo-days useless, UserOrg representatives will likely choose different methods to select the winning bid in future development situations.

5.2 The Relationships Revisited

Influences described in the analysis and illustrated in Figure 1 do not fully explain what happened in the ISD. A closer analysis reveals that the relationship influences often goes via the third element. The following three examples—from DeveloperOrg's perspective—illustrate this finding.

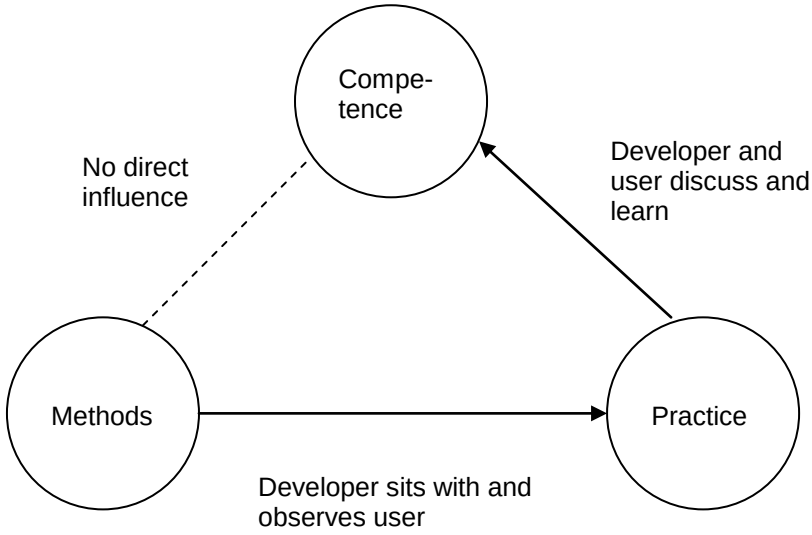

Figure 2: Method influences Competence via Practice.

Method influences Competence via Practice. DeveloperOrg's e-procurement representative chose observation and participation as the method for learning how UserOrg employees use the system to solve daily tasks (see Figure 2). Using this method led to a change in the e-procurement-responsible's domain competence and in system usage.

Competence influences Practice via Method. One of DeveloperOrg's actors had for many years studied how new requirements specifications affect existing systems (see Figure 3). This actor's competence led to his developing and internalizing a method that influenced how he worked in practice to ensure that requirements changes did not adversely affect existing system users.

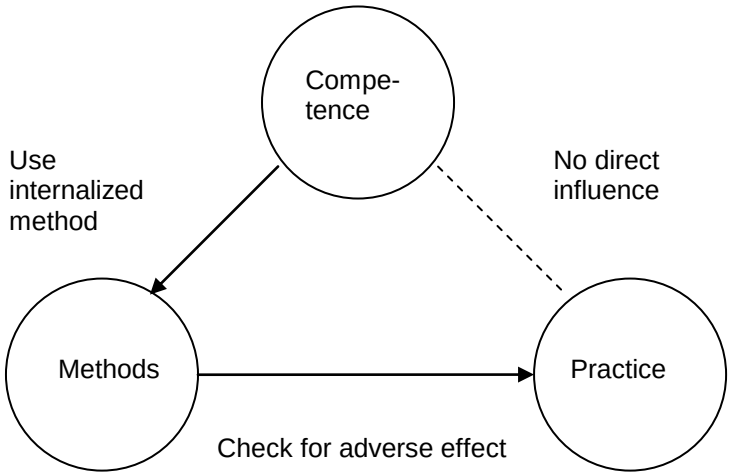

Figure 3: Competence influences Practice via Method.

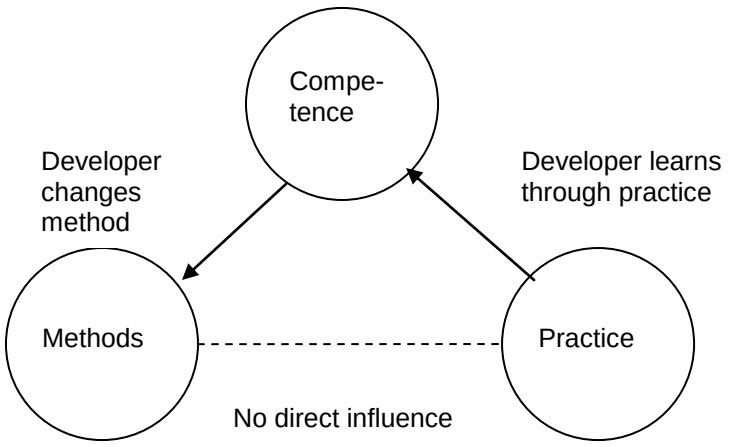

Figure 4: Practice influences Method via Competence.

Practice influenced Method via Competence. DeveloperOrg agreed to develop some special functions for UserOrg. However, uncertainties about the actual usage of such functions within UserOrg led the DeveloperOrg consultant to use his competence to suggest ways (methods) that DeveloperOrg might meet this kind of challenge in future development projects (see Figure 4).

6 Discussion

6.1 Intrinsic Dynamic Relationships

As mentioned in section 5.2, influences between two elements can often be understood or explained only by actively involving the third element. These results both further and add details to Madsen et al.'s (2006) study of the emergent method. The term "development-in-action" is therefore suggested to widen the emergent method's focus to recognize the role of competence and practice, which are at very least as important as methods in an actual dynamic ISD process (Mathiassen and Purao 2002). This change of focus challenges both the emergent method (Madsen et al. 2006) and Fitzgerald et al.'s (2002) Method-in-Action by describing, discussing, and understanding more specifically the influences that occur in ISD's intrinsically dynamic relationships.

Such relationships are dynamic throughout development. This fact is most clearly illustrated in the pilot installation processes, where actors used prototyping as an ISD method to refine requirements specifications and as part of the learning process.

6.2 Common Understanding

The level of commonality between actors' competence, method knowledge, and activities in practice and the clarity of their communications influenced their level of common understanding.

When both DeveloperOrg and UserOrg had high domain competence (ERP development, section 4.3), the difference in development competence did not impact the ISD process. Several factors might explain this. First, the target system was a standard system. DeveloperOrg handled the technicalities, which did not negatively affect UserOrg. The similarities in domain competence made communication easy and clear among actors in both organizations and, in turn, eased the ISD process (Cockburn 2001; Fitzgerald et al. 2002; Mathiassen and Purao 2002).

A change of actors at UserOrg created temporary differences in domain competence (Invoicing module, section 4.3). One actor from each organization shouldered the main responsibilities for continuing this work successfully and increased their domain competence from low to high

because they had good cooperation based on the good relationship they had established earlier in the project.

Different development competence existed between DeveloperOrg (high) and UserOrg (low) throughout the ISD (see section 4.3). DeveloperOrg actors and UserOrg actors did not initially share nor arrive at a common understanding of the ISD method, its use in practice, or the consequences of its use. This caused problems in ISD processes when using prototyping to develop the invoicing module. These problems might be explained as follows. While DeveloperOrg was accustomed to problems with pilot installations, UserOrg did not understand what a "pilot installation" meant. That is, the two organizations had a different understanding of the method's deployment (Ørvik et al. 1999). The DeveloperOrg actors constructed their own representation of what happened (Westrup 1995), and did not heed UserOrg signals that there were problems until UserOrg representatives brought those problems to the attention of DeveloperOrg's top management. In the ISD situation, neither parties' actors understood the importance of the intrinsic relationships and were thus unable to actively clarify the situation before UserOrg escalated it. Ultimately, the problems were resolved through a dialog between top management at both organizations.

As this discussion shows, communication is an imprecise notion. General communication competence is insufficient; specific and shared domain and development competence can help actors obtain a clear and common understanding of what happens in the process (Walz et al. 1993).

As Table 4 shows, negotiation competence played a particular role for DeveloperOrg actors, who used this competence to secure the contract and clarify the requirements specification. This did not negatively influence relationships between the two organizations. As the table also shows, both organizations had project competence, and interview data did not show specific problems related to the technicalities of running the project as such.

6.3 Organizing Vision

While analyzing the case's data, the question of an organizing vision surfaced, inspired by Madsen et al. (2006). They suggest that "organizing around a vision emphasises the need for an IS project to be guided towards a desirable outcome rather than the blind pursuit of a planned result" (p. 236). This case study supports such a suggestion. At the same time, interview data makes this idea problematic in several ways. The organizing vision might be understood differently by different actors (Ørvik et al 1999). Also, the organizing vision might get competition from other visions or goals in the process, or it might change dynamically in the ISD process. Such a change might not be communicated, or might be used tactically by one of the parties to obtain advantages. In addition, the way activities are carried out to reach the vision might clutter the vision, making it difficult for the actors to understand or navigate the processes. For example, both actors might want to develop a good system, but might disagree about what a good system is (as in Fitzgerald et al.'s (2002) discussion of what rationality means for

practitioners vs. formalized methods). Given this, Madsen et al.'s (2006) suggestion might be as challenging to use as a development guide as blindly following a planned result.

As section 4 describes, another example of how challenging it is to reach an organizing vision is visible in the process of eliciting system requirements and creating a common understanding of and agreement on them (Mathiassen et al. 2000). This finding supports the Smolander and Rossi (2008) findings that, when creating an e-business or enterprise architecture in a large, complex ICT company, "the major problems to solve are organizational" (p. 36). Still, it ultimately seems that UserOrg and DeveloperOrg succeeded in agreeing on specifications through a dynamic learning and negotiating process. How can we explain that? Clarity of communication between the actors seems to be the best explanation. They worked together to reach a common understanding of both the situation and the specifications (Cockburn 2001; Walz et al. 1993).

6.4 Implications

Implications for theory. One implication is to emphasize development-in-action rather than focusing on method as in Method-in-Action (Fitzgerald et al. 2002) or the emergent method (Madsen et al. 2006). As Kautz et al. (2007) argue, there are persistent problems and practices in ISD independent of development trends or method use. They propose to focus on dynamic research questions related to diversity, knowledge, social structures, and an understanding of the underlying ISD problems. Research on the intrinsically dynamic relationships in development-in-action could further the understanding of the persistent problems and practices that Kautz et al. (2007) describe.

Implications for practice. In designing and implementing an actual ISD process, it is more important to consider all three elements—competence, methods, and practice—and their intrinsically dynamic relationships rather than focus on methods alone. Both in educating developers and in the reflective systems development processes (Mathiassen 1998), the development-in-action focus can help actors understand, reflect on, and learn ISD processes.

A second implication for practice is that communication is a critical success factor. This is not a new point in the IS field. However, this research suggests that communication challenges in an actual ISD process relate to the degree of commonalities in the actors' competence, methods, and practice; in how they communicate about these factors; and in how they understand the relationships between them. In the prototyping process, for example, this research shows that big differences in competence, methods, and practice can lead to a less successful ISD process.

A third implication is that using a common organizing vision (Madsen et al. 2006) to guide development will be little more than words unless the actors share that vision, understand it in the same way, accept it, and act upon it. Because each actor in an ISD process might have his or her own

agenda in addition to or as part of an organizing vision, the need for clarification is crucial. Understanding and using development-in-action might be one way to achieve such clarification.

7 Conclusion

The main contribution of this study is a deeper and more detailed understanding of the intrinsically dynamic relationships between actors' competence, methods, and practice in an ISD context. The understanding and description of these relationships furthers and details Fitzgerald et al.'s (2002) Method-in-Action and Madsen et al.'s (2006) emergent method and suggests development-in-action as a more suitable term and focus for an ISD process.

How development-in-action emerges in an ISD process depends upon how clearly actors in the process communicate. When actors have common domain and development competence and common organizing visions for the development, the intrinsically dynamic relationships seem to create clear communication and a more successful ISD process.

Recent research shows that large, formalized methods are seldom used in systems development. The research suggests that one reason for this is that formalized methods do not pay enough attention to the individual developer's competence and his or her dynamic use of the method in practice during ISD processes.

This study suggests that it is not only the methods that "emerge" during ISD (Madsen et al. 2006). Both competence and practice also emerge through the interplay between them and the methods deployed in a dynamic ISD process. This emergence calls for further research to clarify what actually takes place, and especially how development-in-action emerges through an ISD process.

Acknowledgements

I am deeply indebted for the challenging and useful comments from Peter Axel Nielsen and the three anonymous reviewers. Thanks also to the "Othello" working group at IRIS'29 led by Lars Mathiassen. Thanks to Bjørn-Erik Munkvold and Tero Päivärinta's for comments on an earlier version of the paper.

References

Avison, D. E., and Fitzgerald, G., (1995). *Information Systems Development: Methodologies, Techniques and Tools*. McGraw-Hill Book Company, London.

Bassellier, G. R., Blaize H., and Benbasat, I., (2001). Information technology competence of business managers: A definition and research model. *Journal of Management Information Systems*, (17:4): 159-182.

Brooks, F., (1987). No silver bullet: essence and accidents of software engineering. *IEEE Computer Magazine, (*April): 10–19.

Ciborra, C.U., (1998). Crisis and Foundation: An inquiry into the Nature and Limits of Models and Methods in the Information Systems Discipline. *Journal of Strategic Information Systems*, (7): 5-16.

Ciborra, C.U., (2002). *The Labyrinths of Information, Challenging the Wisdom of Systems.* Oxford University Press, New York,.

Cockburn, A., (2001). *Agile software development*. Addison-Wesley, Boston, Mass.

Curtis, B., Krasner, H., and Iscoe, N., (1988). A field study of the software design process for large systems. *Communications of the ACM*, (31:11): 1268-1287.

Fitzgerald, B., Russo, N. L., and Stolterman, E., (2002). *Information Systems Development: Methods in Action.* McGraw-Hill Companies.

Glaser, B.G., and Strauss, A.L., (1967). *The discovery of grounded theory: strategies for qualitative research.* Aldine, Chicago.

Huisman, M., and Iivari, J., (2002). The individual deployment of systems development methodologies. *CAiSE 2002*, Springer-Verlag Berlin, pp. 134-150.

Iivari, J., and Hirschheim, R., (1996). Analyzing information systems development: A comparison and analysis of eight development approaches. *Information Systems Journal*, (21:7): 551-575.

Iivari, J., and Lyytinen, K., (1998). Research on Information Systems Development in Scandinavia – Unity in Plurality. *Scandinavian Journal of Information Systems*, (10:1&2): 135-186.

Jayaratna, N., (1994). *Understanding and evaluating methodologies. NIMSAD: A Systemic Framework.* McGraw-Hill Book Company.

Kautz, K., Madsen S., Nørbjerg, J., (2007). Persistent Problems and Practices in Information Systems Development. *Information System Journal*, (17:3): 217-239.

Kiely, G., and Fitzgerald, B., (2003). An investigation of the use of methods within information systems development projects. In: *Proceedings of IFIP WG 8.2 Conference, Athens Greece*.

Kraiger, K., Ford, K., and Salas, E., (1993). Application of cognitive, skill-based and affective theories of learning outcomes to new methods of training evaluation. *Journal of Applied Psychology*, (78:2): 311-328.

Lee, D. M. S., Trauth, E. M., and Farwell, D., (1995). Critical skills and knowledge requirements of IS professionals: A joint academic/industry investigation. *MIS Quarterly*, (19:3): 313 - 340.

Madsen, S., Kautz, K., and Vidgen, R., (2006). A framework for understanding how a unique and local development method emerges in practice. *European Journal of Information Systems*, (15): 225-238.

Mathiassen, L., (1998). Reflective Systems Development. *Scandinavian Journal of Information Systems*, (10:1&2): 67-118.

Mathiassen, L., Munk-Madsen, A., Nielsen, P. A., and Stage, J., (2000). *Object-Oriented Analysis and Design*. Forlaget Marko, Aalborg.

Mathiassen, L., and Purao, S., (2002). Educating reflective systems developers. *Information Systems Journal*, (12): 81-102.

Marcolin, B. L., Compeau, D, R., Munro, M. C., and Hupp, S. L., (2000). Assessing user competence: Conceptualization and measurement. *Information Systems Research*, (11:1): 37-60.

Necco, C. R., Gordon, C. L., and Tsai, N. W., (1987). Systems Analysis and Design: Current Practices. *MIS Quarterly*, (11:4): 461-478.

Schaumbach, T.P., (1994). Maintaining professional competence: an evaluation of factors affecting professional obsolescence of information technology professionals. Ph D dissertation. *University of South Florida*.

Smolander, K., and Rossi, M., (2008). Conflicts, Compromises, and Political Decisions - Methodological Challenges of Enterprise - Wide E-Business Architecture Creation. *Journal of Database Management*, (19:1): 19-40.

Truex, D., Baskerville, R. and Travis, J., (2000). Amethodical systems development: the deferred meaning of systems development methods. *Accounting Management and Information Technologies*, (10:1): 53-79.

Walz, D.B., Elam, J.J. and Curtis, B., (1993). Inside a Software Design Team: Knowledge Acquisition, Sharing, and Integration. *Communication of the ACM*, (36:10): 63-77.

Wastell, D., (1995) The fetish of technique: methodology as a social defense. *Information Systems Journal*, (6:1): 25-40.

Westrup, C., (1995). *Transforming Organizations Through Systems Analysis: Deploying New Techniques for Organizational Analysis in IS Development.* IFIP WG8.s Working Conference on Information Technology and Changes in Organizational Work,.

White, K. B., and Leifer, R., (1986). Information systems development success. Perspectives from project team participants. *MIS Quarterly*, (10:3*)*: 214-223.

Yin, R. K., (1994). *Case study research: design and methods* (2nd ed.). Sage Publications.

Zalkowitz, M., Yeh, R., Hamlet, R., Gannon, J., and Basili, V., (1984). Software engineering practices in the U.S. and Japan. *IEEE Comput*., (17:6): 57-66

Ørvik, T. U., Olsen, Dag H., and Sein, M. K., (1999). Deployment of System Development Methods. Exploring Paradigmatic Mismatches. Evolution and Challenges. In: *System Development*, Zupancic et al. eds., Kluwer Academic/Plenum Publishers, New York, pp. 19-31.

Actors' Competencies or Methods? A Case Study of Successful Information Systems Development

Hans Olav Omland[1] and Peter Axel Nielsen[1,2]
[1]University of Agder, Kristiansand, Norway
[2]Aalborg University, Aalborg, Denmark
hans.o.omland@uia.no, pan@cs.aau.dk

Abstract

Research on information systems development methods has by and large acknowledged a significant difference between a method and its use and that the use depends on the situation and the developers, as well as other contingencies. We extend this research and focus in particular on the relationship between actors' competencies and their deployment of methods, arguing that this relationship is described over-simplistically and needs a better explanation. Through a case study of a successful information systems development project we identify some central situations where a variety of competencies and methods are exercised. Emphasising the intertwining of competencies and methods, we discuss the character of the intertwining process, how different actors relate to different methods, and how methods may be part of the problem rather than part of the solution to challenges in information systems development. The paper suggests elements for a new model for explaining actors' competencies and their use of methods.

Keywords
Competence, methods, actors, information systems development, analysis.

INTRODUCTION

In the research literature on methods for information systems development (ISD) there is an increasing awareness that the learning and use of methods is a social activity that cannot be understood simply by studying methods' features. In this paper we will add to this view by addressing the relationship between actors' competencies as exercised in ISD projects on the one hand and their use of methods on the other hand. Before we state our detailed research question we will present some of the recent research results relevant to the paper's focus.

A recent compilation of methods is that by Avison and Fitzgerald (2003) where several methods are presented and their features are discussed. Methods' features and their philosophical foundations are discussed in even more detail in Hirschheim et al. (1995). Research identifying and describing methods' features is vast: distinctions between methods (Nielsen 1989), feature analysis (Olle et al. 1983; 1986; Iivari 1994), paradigmatic analyses (Iivari et al. 1998; Russo and Stolterman 2000), processes for requirements engineering (Sommerville and Kotonya 1998), combinations of methods (Avison 1990; 1997; 1998; Vidgen 2002). The pure feature analyses have been criticised for not taking developers' practice and their situational contingencies into consideration when evaluating methods (Nielsen 1991). Much research has thus been directed at evaluating methods (e.g., Jayaratna 1994; Nielsen 1991; Siau and Rossi 1998; Siau et al. 1996; 1997). Siau and Tan (2005) make the point that evaluation criteria are still under-developed and they elaborate their own criteria extensively. Most of this research evaluates methods in the context of their use and hence seeks to relate features of methods to particular contingencies or at least explain how methods and situations can be matched given a particular situation.

Critique has been directed at the instrumental view of methods whereby methods are seen as 'fitting' particular situations. It is argued that the practice of ISD does not follow methods, and it is a-methodical (Truex et al. 2000). Others follow a similar view and have found through empirical studies that methods emerge through practice (Madsen et al. 2006), that methods are never used by-the-book (Fitzgerald 1997; Kiely and Fitzgerald 2003), or that it is necessary to have a critical view on the use of methods (Fitzgerald 1996).

Another strand of research on methods has been concerned with tailorising methods to the unique situation in which they are to be used. Nielsen (1991) proposes that this should be based on a soft systems analysis of the development task. Harmsen et al. (1994) and Brinkkemper (1996) suggest that methods are engineered on the spot to fit the

current situation and needs. Reports from empirical studies show how this has been done in practice at Motorola (Fitzgerald et al. 2003) and at Intel (Fitzgerald et al. 2006).

In most of the research on ISD methods, the actors, i.e. people using the methods, seem to be absent. There are exceptions, however. Nielsen (1991) identified different ways in which developers use methods depending on their experience and values. The more experienced developers perceived methods as less necessary. The more developers possessed values of the profession, e.g. systematic approach, reflective attitude, education, code of ethics, the more methods were appreciated (though never used blindly). Fitzgerald et al. (2002) acknowledge the importance of developers in their model of method-in-action. In their model they take a broad view on developers and include all involved actors and hence also stakeholders in the use organisation. In their model they let developers enact methods-in-action and make it clear that developers have different skill levels. Developers have capabilities, learn over time, have knowledge of the application domain, and have some degree of autonomy, commit, and exercise personal motivation (Fitzgerald et al. 2002, p. 123-134). Though they have a concern for the actors' use of methods, they limit themselves to developers, leaving out other actors.

In this paper we will take the view that it is important to understand the developers' use of methods, but we will supplement this with other actors who are also involved in the use of methods. We find the relationship between users and developers to be important in general in ISD. The reasons are many (Nandhakumar and Jones 1997): it improves the requirements process and the design process, it furthers organisational implementation, and it furthers ethical principles, and we may add that it furthers workplace democracy (e.g., Bjerknes and Bratteteig 1995). Not only will we need to look at developers, users, and other stakeholders whom we will refer to simply as actors, but we will also focus our attention on the actors' competencies. We will do this for the same reason as Fitzgerald et al. (2002), i.e. that methods are enacted through the actors and that their competencies will probably influence how they use methods and how they perceive methods (Nielsen 1991, p. 73; Orvik et al. 1999).

The research focus in this paper is thus: how do actors' competencies and their use of methods relate? and to the extent that the process of exercising actors' competencies and the process of using methods are intertwined, how do the two processes influence each other?

In the next section we present what we take actors' competencies to be and how we see ISD methods. Thereafter we explain our research approach as a case study, emphasising data collection and data analysis. Then follows a section with a case description in which we give a short description of the case followed by a section with the case analysis where we use episodes and encounters inspired by Robey and Newman (1996). After the case analysis follows discussion in which we draw attention to the implications of our findings and in particular we assess how different actors use different methods and how the methods may become part of the problem rather than part of the solution. The paper then concludes.

BACKGROUND

'What is Competence?' Le Deist and Winterton (2005) research differences and similarities between understanding of competence in the US, UK, France, Germany, and Austria. From their analysis they argue that a holistic typology is useful in understanding the combination of knowledge, skills, and social competencies that are necessary for particular occupations. They categorise competence in two dimensions: occupational/personal and conceptual/operational (Figure 1). Le Deist and Winterton (2005) use the term 'meta competence' to describe the competence to reflect (including learning to learn) in order to facilitate the acquisition of the other competencies.

	Occupational	**Personal**
Conceptual	Cognitive Competence	Meta competence
Operational	Functional Competence	Social competence

Figure 1. Typology of competence (Le Deist and Winterton 2005)

According to Le Deist and Winterton (2005) and other writers a general typology of competence is knowledge, skills and attitude. In their topology 'cognitive competence' captures knowledge and understanding, 'functional competence' equals skills, and 'social competence' includes behavioural and attitudinal aspects. The typologies in the IS field are basically the same as the one described by Le Deist and Winterton (2005). Lee et al. (1995) investigated critical skills and knowledge requirements of IS professionals and found that industry demands IS professionals with knowledge and skills in the areas of technology, business operations and management, and interpersonal skills to lead organisational integration and process re-engineering activities effectively. Feeny and

Willcocks (1998) suggest that skills are divided into three different areas: business, technical and interpersonal. White and Leifer (1986) define the top five competencies needed by project team members as: business knowledge, good communication skills, technical skills, analytical skills and good organisational skills. Peppard et al. (2000) surveying the IS research literature observe that the competence in the IS literature is predominantly focused upon the individual in the form of IS skill sets. Bassellier et al. (2001) quote Nordhaug (1998) in that competence at the individual level is required for the creation of core competence at the organisational level.

In the IS field competence is, generally speaking, 'the ability to…', (Peppard et al. 2000, p. 302). Competence is the ability, the enabler, providing the means for performance (Bassellier et al. 2001). According to Bassellier et al. (2001), competence and performance cannot be used interchangeably as that will lead to confusion. They refer to Schambach (1994) and state that the terms are related, but 'factors other than competence – such as motivation, effort, and supporting conditions – may influence performance' (p. 162). Performance is dependent on motivational values and these values seem to change over time, leading to changing interests and areas of performance for the IS professionals (Feeny and Willcocks 1998).

Sandberg (2000) researches human competence at work and challenges Bassellier et al.'s (2001) views. He discusses the prevalent rationalistic approaches where 'human competence at work is seen as constituted by a specific set of attributes, such as the knowledge and skills used in performing particular work' (p. 9). He is referring to interpretative research and says that the person and the world are inextricably related through the person's lived experience of the world, stating that competence is not seen as consisting of two separate entities. 'Instead, worker and work form one entity through the lived experience of work' (Sandberg 2000, p. 11). Hager and Gonczi (1996) argue that much of the existing view of competence is too atomistic and suggest an integrated approach. Forgetting about attributes and concentrating on tasks is, according to Hager and Gonczi (1996), the primary reason why so many people lapse into a narrow view of competence standards, meaning that competence standards are often thought of and approached as simply a series of discrete task descriptions. The integrated conception of competence is contextualised by selecting key tasks or elements that are central to the practice of the profession it relates to. This means that Sandberg's (2000) 'worker and work form one entity' (p. 11) points to 'competence-in-action' is an alternative to just competence as "the ability to…" (Peppard et al. 2000, p. 302).

Jayaratna (1994) stated that there were over one thousand definitions of the 'ISD method'. This figure is possibly much higher today. Fitzgerald et al. (2002) define method as 'a coherent and systematic approach, based on a particular philosophy of systems development, which will guide developers on what steps to take, how these steps should be performed and why these steps are important in the development of an information system' (p. 5). This definition includes both a conceptual and a philosophical basis and some practical actions that lead to a coherent and systematic approach in ISD. Fitzgerald et al. (2002) use the term formalised methods to refer to commercial, brand-named methods and include internally developed and formally documented methods in the same term.

In spite of all the different definitions and descriptions of methods it appears that methods are not used fully. It seems that the construction of methods is a more popular activity than their actual use. Developers may use their own 'in-house' methods or contextualising methods in specific situations (Kiely and Fitzgerald 2003). The method user may understand, interpret and contextualise the method differently (Orvik et al. 1999).

'Methods are supposed to change, and ideally, improve practice. Methods are used because the established work culture does not deliver results in a desirable fashion' (Fitzgerald et al. 2002, p. 9). Still, ISD projects seem to be difficult to control as many projects overrun on both cost and time, and deliver less functionality than initially planned. Fitzgerald et al. (2002) state that the primary reason for their research is to describe and clarify the tension between formalised methods and method-in-action. They assume that 'better use of methods can be achieved through a more comprehensive understanding of this tension' (p. 8). Madsen et al. (2006) state that method-in-action builds on a static conception of reality. They use the term 'emergent method' to describe the dynamics that take place in development projects.

RESEARCH APPROACH

This research adopts an exploratory case study approach since it seeks to answer 'How' questions (Yin 1994), questions about the relationships between actors' competence and their use of methods. This is done through 'an empirical inquiry that investigates a contemporary phenomenon within its real-life context' (Yin 1994 p. 13) where 'the investigator has little control over events' (Yin 1994 p. 1).

This case study is based on a systems development project where a large local municipality in Norway needed a new ERP system with several new functions. The empirical data were mainly collected through semi-structured interviews and study of project documentation. Fifteen people were interviewed. Six interviewees were from the developer company, including the project manager, the product managers responsible for the ERP system, the invoicing system, the e-procurement system, and the domain experts engaged in the project. Nine interviewees were from the municipality including their project manager, the project coordinator, and the sub-project managers. The interviews were conducted in retrospect after the main part of the project was implemented. They were tape recorded and transcribed. The transcriptions were sent to each of the interviewees for validation. Feedback on the transcribed interviews was received via e-mail.

The analysis of the data was inspired by grounded theory (Glaser and Strauss 1967) in the following way. The transcribed interviews were subjected to open coding based on the seed categories 'actors' competence' and 'methods', and were therefore less open. The code 'method' was used in the sense of parts of methods like prototyping. The interviews were also subjected to axial coding as the aim of the research was to understand the relationships between the seed categories. Finally, the findings were written into two reports and sent to the customer and the vendor respectively for validation. A follow-up interview was then conducted with the municipality's project manager to get feedback on the report. Feedback was also received through e-mail both from the municipality and the developer company. Based on this information and the axial coding the relationships between competence and methods were explained by the use of meta competence and the goals pursued by the different actors.

The findings were then organised into 'encounters' and 'episodes'. Robey and Newman (1996) describe encounters as concentrated events carrying 'opportunities to address prior performance, to express dissatisfaction, and to plan for meeting future needs' (p. 33). Episodes are described as events of 'relatively long periods of equilibrium' (p. 33).

CASE DESCRIPTION

The municipality were searching for a new ERP system since their previous provider announced that they would stop supporting the municipality's existing ERP system. Two development companies competed to win the contract. The winner had to develop and modify their initial system to fulfil the municipality's requirements.

Course of actions

The ERP system included the following modules: personnel and salary, accounting, invoicing citizens in the municipality, budget and e-commerce. The different departments in the municipality started their development process by brainstorming and discussing what they wanted from the new system. Their findings were published internally in the departments and discussed until consensus was reached. Afterwards the different sub-project managers met with the overall project manager and the project coordinator in the municipality. The project coordinator modelled the business processes of the different departments through what was called 'a process-oriented tendering process' (municipality's project coordinator), ending with a complete tender document. Two competing development companies had to demonstrate their systems based on a case made by the municipality. The purpose was: '... to expose the weaknesses in the system and what they [the development companies] could deliver' (municipality's project manager). After further clarification the two development companies submitted their final and best bids. After the contract was signed the municipality and the winning development company continued to clarify the requirements of the new system. The development project is reported by both the municipality and the developer company to be successful, at least based on delivery, budget, functions and use. The development started in the municipality in 2001 and was finished in spring 2005 when all modules of the ERP system had been installed. The developer used parts of methods, but no particular and complete systems development method.

The developer company was eager to get the municipality as their customer for all parts of their ERP system. The developer company took this opportunity to upgrade their existing system, redevelop some modules and develop several new modules in close cooperation with the municipality. The developer company decided early in the process that they would win the contract at all costs.

Encounters and episodes

Table 1 describes the development project by encounters and episodes. It also describes competencies used, methods used and the goals of the actors participating in the development. The episodes and encounters are listed chronologically. In Table 1 'domain competence' is competence in the actual working practices, and 'business processes' competence encompasses the processes that were used to achieve the expected results in daily work.

Prior to the first encounter between the municipality and the developer companies the municipality's goal was to do a simple swap of ERP systems leaving their business processes untouched.

Table 1. Project activities related to competencies used, methods used and the goals the actors wanted to reach; M=municipality; D=developer company.

En/Ep	Activities (driver)	Competencies used (involved)	Method used (involved)	Goal (involved)
En1	Initiate project (M)	Project (M)		Swap system (M)
Ep1	Elicit requirements (M)	Domain (M), Project (M), Business processes (M)	Brainstorming (M) Document study (M)	Requirements for new system (M)
En2	Clarify require-ments (M)	Domain, Project (M)	Modeling (M)	Specification/bid document (M)
Ep2	Prepare systems presentations (M)	Domain (M) Business Processes (M)		Test developer organisations and their systems (M)
En3	Systems presentations (D,M)	Domain (D, M) Technical (D) Presentation (D) Business processes (D, M)	Demonstrate system (D) Testing (M)	Find how system fits, and customize (M) Win the contract (D)
Ep3	Prepare the bid (D)	Domain, technical and bid/contractual (D)	Evaluating (D) Prepare documents (D)	Win the contract (D) Improve system (D)
En4	Bid delivered (D)			Win the contract, upgrade own system (D)
Ep4	Clarify the bid (D, M)	Domain (D, M) Technical (D) System functions and use (D) Business processes (D, M)	Test (M) Communication (D, M) Evaluation (M)	Check system fit and further development (M) Sell system and improve it (D)
En5	Best bid (D)			Win the contract (D)
Ep5	Consider the bid (M)	Domain (M) Negotiation and strategy (D) Business processes (D, M)	Comparison (M) Legal advice	Documentation for decision on best system functions and price (M)
En6	Contract (D, M)			Best system (D, M)
Ep6	Develop and implement (D, M)	Domain (D, M) Relationship (D) Communication (D, M) Development (D) Business processes (D, M)	Negotiations (D, M) Testing (M) Strategic planning (D) Prototyping (D) Parametric customise (D)	Functional system (M) Upgrade system (D) Win municipality as customer and consider existing customers (D)
En7	Meeting at a high level (M)	Development (D) Contractual Meta (D, M)	Meeting (D, M)	Solve problems and set project on track (D, M)
Ep7	Continue Ep6 (D,M)	Continue as in Ep 6	Continue as in Ep 6	Continue as in Episode 6
En8	Instal main modules (D, M)	Test (M) Business processes (D, M)	Test run (M)	Perform daily tasks (M) Improved system (D)

CASE ANALYSIS

The jointly driven encounters and episodes gave more room for the interactions between the different actors. Use of the different competencies and methods will therefore be described and discussed in more detail.

Encounter 3: Systems presentation. Two developer companies were invited to present their ERP systems based on a demo case made by the municipality. The presentation was a frustrating experience for both parties as neither of them fully reached their goals. Different goals and the lack of discipline in following the demo case seem to be part of the explanation. The municipality's actors asked many questions, leaving the developer companies' presenters with an unfinished presentation. The systems manager at the developer company said that the demo case did not give them opportunities to present important aspects of their system. Even if both parties had communication competence it seemed that they were not able to communicate clearly. One reason may be lack of reflection, i.e. not using meta competence, another may be that the developer company wanted to keep a good relationship with the municipality as the buyer, and a third that the two parties simply did not understand that the communication was unclear.

Episode 4: Clarify the bid. This was an intensive process between the two parties where the developer company acted strategically to improve their system and not create unnecessary problems for the existing system users.

Episode 6: Develop and implement. When the development started the requirement specifications were still not finalised. For the standard modules this proved to be no problem as many functions were standard functions and could be parametrically customised to the municipality, and the development had no negative influences on the user.

For the invoicing module the requirement specification was discussed and dynamically elicited during the project. The municipality's domain competence in this area was on a high, national level and the developer company were eager to profit from this competence to improve their system. One of the major conflicts in the project developed during this episode. The developer company used prototyping to elicit requirements and to develop the system. The municipality did not have prototyping competence and probably did not understand that they were pilot users of the system. All the actors had demonstrated communication competence, but it seemed that the many errors in the prototype and the municipality's assumption that the system should be more complete than it was actually inflamed the conflict between the two parties. One of the developer company's actors said that the municipality was a pilot user, but wondered if the municipality had understood that.

Encounter 7: High-level meeting. After pressure from the municipality a high-level meeting was called between both parties to sort out the problems and agree on a more organised ISD process and better planned releases of modules. This resolved the frustration and enabled the municipality to test releases and give feedback to the developers.

Episode 7: Develop and implement. This episode was a continuation of episode 6 with the new antecedent conditions mentioned in encounter 7. The developer company changed their release policy to allow the municipality to test and give feedback on their ideas for further development of the system.

Encounter 8: Instal main modules. The main modules were installed at the beginning of 2004, conveniently following the fiscal year. The invoicing module was installed by the end of 2004, giving more time for the development. The systems development is regarded as successful. The system is presently used by the municipality.

Competencies deployed

Guided by Peppard et al.'s (2000) definition of competence as 'the ability to..' (p. 302) we analysed the interview data and inferred from them what competencies were deployed. We found that all types of competence described by Le Deist and Winterton (2005) were deployed (see Table 2). The meta competence was most visibly deployed to solve the conflict described between the developer and the municipality.

Further analysis of Encounter 7

In episode 6 two situations arose: the parametric customisation of the system and the development of the invoicing module. These two situations were similar in all aspects except for two: the major method deployed and the consequences of the method deployment for the municipality's actors. The differences in method competence between the developer company and the municipality were basically the same in the two situations. The big difference was the consequences of method deployment for the municipality. Parametric customisation was handled successfully and the municipality did not have any negative experiences with the method of deployment. During the development of the invoicing module the municipality experienced many negative consequences, leading them to push for a high-level meeting in encounter 7.

How can we explain this difference? For the development of the invoicing module the municipality's actors did not understand that they were pilot users and that prototyping was used in the development. They complained and made a lot of fuss during the development. The developers and their project managers had competence in a prototyping method and were used to problems during prototyping. They therefore did not respond properly to the municipality's complaints. Even if both parties had communication competence and a good relationship at the beginning of the development project they were unable to communicate and resolve this situation before it escalated. Their competence and experience in deploying the prototyping method seemed to hinder the developers from understanding that the problems were escalating. In this respect the method and its deployment became part of the problem.

In encounter 7 managers from the developer and the municipality met, deployed meta competence when reflecting on what had happened and identified reasons for the misunderstandings and heated arguments between the actors. They then agreed on ways to proceed to get the development process back on track again. Even if cognitive, functional and social competence were present in episode 6 the actors did have to use some meta competence to reflect on their own deployment of the mentioned competencies and make changes that the different actors accepted.

Goals

In the successful part of the development project the domain where the system was to be used was well established and the different actors worked towards the same goals, or else the goals were not visibly in conflict. For the invoicing module, strategic changes in the goals surfaced as one factor explaining what happened. The different actors viewed the goals from different perspectives and acted strategically in reaching their goals according to their own understanding as exemplified in the requirement elicitation. The municipality wanted special functions to become standard in the system whereas the developer company had to consider the system holistically and how new functions would influence the existing and other customers. The developer company's strategy proved successful. The municipality changed its strategy from just wanting to swap systems to actively participating in the development of the new system and system modules even if some of the actors initially resisted changing the ERP system.

Table 2. Classification of competencies based on Le Deist and Winterton (2005)

Type	**Competencies**	Developer company					Municipality			
		Board level	Proj. mgr.	System mgr.	Developers	Program-mers	Proj. mgr.	Proj. grp.	Sub proj. mgr.	Proj. coor.
Cognitive	Domain			√	√			√		
	Strategic	√	√	√						
	Business process			√	√			√	√	
Operational	General dev			√	√	√				
	Prototyping			√	√					
	Project		√	√	√		√		√	√
	Presentation			√	√					
	Systems function and use			√	√		√		√	√
	Technical			√	√					
Meta	Reflecting and making changes	√	√	√			√		√	
Social	Relationship				√					
	Communication	√	√	√	√	√	√	√	√	√

Methods part of both solutions and problems?

Methods deployment seems to be part of both successes and problems in this development. Success or problem depends on the development project itself, the consequences for the different actors, how the actors understood the situation and their ability to handle the situations arising during the development project. Applying meta competence may enable the actors to reflect on their experiences and devise ways of solving problems before the problems escalate or sort them out after the development situation has escalated and thereby further the ISD process successfully. Use of meta competence or rather lack of using it was especially visible where the developer and their managers did not take care of the negative consequences their method of deployment had for the municipality's actors. They did not react to the signals from the municipality's actors before the municipality brought the situation to the attention of high-level managers.

DISCUSSION

The above analysis points to meta competence (Le Deist and Winterton 2005) as an important aspect in handling challenging parts of development projects. In this context meta competence means the ability to reflect on and learn from the results of deploying the other three types of competence. This expands the traditional understanding of competence in the IS field as business, technical and interpersonal competence (Feeny and Willcocks 1998). In situations where method use leads to negative consequences for some actors meta competence may be critical to solve the problems and continue the development project as communication competence itself may not be enough.

In the encounters and episodes discussed, the actors' competence, methods and activities are intertwined making it difficult to point clearly to what is method and what is competence in the dynamic ISD processes. An example of the intertwining is seen in the developer's handling of the requirements elicitation. The developer company derived requirements together with the municipality through discussions and tests. At the same time the developer company

had to make sure that the requirements did not infringe on the existing customers. In this process it was difficult to state what was method use and what was competence deployment. This supports Sandberg's (2000) description of the relationship between worker and work as forming 'one entity through the lived experience of work' (p. 11). This relationship is not yet researched in the IS field. Madsen et al.'s (2006) focus on the emergent method does not fully integrate these aspects. The findings therefore suggest that Madsen et al.'s concept of emerging method needs to be expanded to include the understanding of the intertwining between all the actors' deployment of competence and method in their activities. This is particularly relevant as different actors may understand and deploy the same methods in different ways (Orvik et al. 1999). Generally speaking, and specifically relating to Fitzgerald et al. (2002) and Madsen et al. (2006), the IS field lacks the understanding of what Sandberg (2000) calls 'one entity' and what we have explained here as the intertwining of competencies and use of methods.

Pursuit of goals in the development process may be complicated by the lack of understanding of the intertwinedness between competencies and methods. The actors may seemingly pursue the same goals, but without deploying some kind of meta competence they may not be able to understand their own and other actors' goals. This research provides a more detailed understanding of Madsen et al.'s (2006, p. 226) 'unfolding of the actual development process as an outcome of a complex web and interplay of enacting and interacting actors and structures' and suggests that the ISD process including the intertwinedness of competence and methods is a more important subject for study than the 'emergent method'. This is particularly relevant since the different actors' goals or vision for the system may change depending on the possibilities or constraints different actors meet in the development process.

The above discussion shows that it is important to understand the development process as a whole and that it is influenced by the competencies deployed, the methods used and the goals pursued by all the participating actors. These are also mutually influencing each other and they are influencing and being influenced by the system being developed. Failure to recognise and take these influences into consideration may lead to the method becoming part of the problem, as exemplified in this case, and thereby not part of the solution. The research suggests further that the traditional understanding of competencies within the IS field needs to be expanded by the meta competence enabling all the actors to reflect on the development processes and hence improve processes while in the midst of action instead of letting conflicts build unnoticed.

This research therefore suggests that a major activity in designing and carrying out contextualised ISD processes is to consider how the methods and all the actors' competencies present in the development team can make a good fit dynamically from start to finish through the deployment of meta competence. This should also be considered by method designers.

CONCLUSION

The research focus in this paper has been: how do actors' competencies and their use of methods relate? and to the extent that the process of exercising actors' competencies and the process of using methods are intertwined, how do the two processes influence each other? Influences and intertwinedness between competencies and method depend on the actual situation, the goals pursued by the individual actors, and the use of meta competence. The research suggests that competencies and methods deployed in a given development situation are intertwined in such a way that they cannot be separated in practice. This is in line with Sandberg (2000) and is not well researched in the IS field. It is therefore a contribution to the understanding and application of ISD methods.

In this case study the clearest difference between success and failure is seen where the developer company's actors had prototyping competence and deployed prototyping as a method without using meta competence in reflection over the consequences for and reactions from the municipality's actors. The municipality lacked both prototyping competence and understanding of prototyping as a method.

The goals pursued by the different actors in the development were important for the intertwining of competencies and methods, and were also part of the conflicts where goals conflicted or changed dynamically during the development process. This research therefore suggests that a new model for ISD needs to focus on the ISD process as a whole including the competencies deployed, the method used, the goals pursued and all the actors involved.

REFERENCES

Avison, D. and Wood-Harper, T. 1990. *Multiview Methodology*. Oxford: Blackwell Scientific Publishers.

Avison, D.E. and Wood-Harper, A.T. 1997. *Multiview*. Maidenhead, UK: Alfred Waller.

Avison, D.E., Wood-Harper, A.T., Vidgen, R.T. and Wood, J.R.G. 1998. "A further exploration into information systems development: the evolution of Multiview 2," *Information Technology & People* (11:2), pp 124-139.

Avison, D. E., and Fitzgerald, G. 2003. "Where now for development methodologies?," *Communications of the ACM* (46:1), January, pp 78-82.

Bassellier, G., Reich, B. H., and Benbasat, I. 2001. "Information Technology Competence of Business Managers: A Definition and Research Model," *Journal of Management Information Systems* (17:4), Spring, pp 159-182.

Bjerknes, G., and Bratteteig, T. 1995. "User Participation and Democracy: A Discussion of Scandinavian Research on System Development," *Scandinavian Journal of Information Systems* (7:1), pp 73-98.

Brinkkemper, S. 1996. "Method engineering: engineering of information systems development methods and tools," *Information and Software Technology* (38:4), pp 275-280.

Feeny, D. F., and Willcocks, L. P. 1998. "Core IS Capabilities for Exploiting Information Technology," *Sloan Management Review* (39:3), Spring, pp 9-21.

Fitzgerald, B. 1996. "Formalized systems development methodologies: A critical perspective," *Information Systems Journal* (6:1), pp 3-23.

Fitzgerald, B. 1997. "The use of systems development methodologies in practice: a field study," *Information Systems Journal* (7:3), pp 201-212.

Fitzgerald, B., Russo, N. L., and Stolterman, E. 2002. *Information Systems Development: Methods in Action*. McGraw-Hill Companies.

Fitzgerald, B., Russo, N.L., and O'Kane, T. 2003. "Software Development Method Tailoring at Motorola," *Communications of the ACM* (46:4), April, pp 65-70.

Fitzgerald, B., Hartnett, G. and Conboy, K. 2006. "Customising agile methods to software practices at Intel Shannon," *European Journal of Information Systems* (15:2), pp 200–213.

Glaser, B. G., and Strauss, A. L. 1967. *The discovery of grounded theory: strategies for qualitative research.* Chicago: Aldine.

Hager, P., and Gonczi, A. 1996. "What is competence?," *Medical Teacher* (18:1), pp 15-18

Harmsen, F., Brinkkemper S., and Oei, H. 1994. "Situational Method Engineering for Information System Project Approaches," *IFIP Transactions Vol. A-55*, pp 169-194. Amsterdam: North-Holland,

Hirschheim, R. A., Klein, H.-K., and Lyytinen, K. 1995. *Information systems development and data modelling: conceptual and philosophical foundations*. Cambridge: Cambridge University Press.

Iivari, J., Hirschheim, R., and Klein, H. K. 1998. "A paradigmatic analysis contrasting information systems development approaches and methodologies," *Information Systems Research* (9:2), June, pp 164-193.

Iivari, J. 1994. "Object-oriented information systems analysis: Comparative analysis of six object oriented methods," *IFIP Transactions Vol. A-55*, pp 85-110

Jayaratna, N. 1994. *Understanding and evaluating methodologies: NIMSAD: a systematic framework*. London: McGraw-Hill.

Kiely, G., and Fitzgerald, B. 2003. "An investigation of the use of methods within information systems development projects," *IFIP WG 8.2*, Athens Greece.

Le Deist, F. o. D., and Winterton, J. 2005. "What Is Competence?," *Human Resource Development International* (8:1), March, pp 27-46.

Lee, D. M. S., Trauth, E. M., and Farwell, D. 1995. "Critical Skills and Knowledge Requirements of IS Professionals: A Joint Academic/Industry Investigation," *MIS Quarterly* (19:3), September, pp 313-340.

Madsen, S., Kautz, K., and Vidgen, R. 2006. "A framework for understanding how a unique and local development method emerges in practice," *European Journal of Information Systems* (15), pp 225-238.

Nandhakumar, J., and Jones, M. 1997. "Designing in the dark: the changing user-developer relationship in information systems development," *Proceedings of the eight ICIS*, Atlanta, Georgia, pp 75-88,

Nielsen, P. A. 1989. "Reflections on Development Methods for Information Systems A Set of Distinctions Between Methods," *Office: Technology & People* (5:2), pp 81-104.

Nielsen, P. 1991. *Learning and Using Methodologies in Information Systems Analysis and Design*. Ph D thesis Lancaster University.

Nordhaug, O. 1998. "Competence Specificities in Organizations," *International Studies of Management & Organization* (28:1), Spring, pp 8-29.

Olle, T.W., Sol, H.G. and Tully, C.J. (Eds.) 1983. *Information Systems Design Methodologies: A Feature Analysis*. Amsterdam, Netherlands: Elsevier Sci. Publications.

Olle, T.W., Sol, H.G., Tully, C.J. and Verrijn-Stuart, A.A. (Eds.) 1986. *Information systems Design Methodologies: A Comparative Review*. Amsterdam: New-Holland

Orvik, T. U., Olsen, D. H., and Sein, M. K. 1999. "Deployment of System Development Methods. Exploring Paradigmatic Mismatches," In Zupancic, J. et al. (Eds.), *Evolution and Challenges in System Development*, pp 19-31. New York: Kluwer Academic/Plenum Publishers.

Peppard, J., Lambert, R., and Edwards, C. 2000. "Whose job is it anyway?: organizational information competencies for value creation," *Information Systems Journal* (10:4), October, pp 291-322.

Robey, D., and Newman, M. 1996. "Sequential Patterns in Information Systems Development: An Application of a Social Process Model," *ACM Transactions on Information Systems* (14:1), January, pp 30-63.

Russo, N. L., and Stolterman, E. 2000. "Exploring the assumptions underlying information systems methodologies," *Information Technology & People* (13:4), pp 313-327.

Sandberg, J. R. 2000. "Understanding Human Competence at Work: An Interpretative Approach," *Academy of Management Journal* (43:1), pp 9-25.

Schambach, T. P. 1994. *Maintaining professional competence: an evaluation of factors affecting professional obsolescence of information technology professionals*. Ph. D. dissertation, University of South California.

Siau, K., Wand, Y. and Benbasat, I. 1996. "Evaluating information modelling methods - A cognitive perspective," *Proceedings of the 1st Workshop on Evaluation of Modelling Methods in Systems Analysis and Design*, Crete, Greece, May 20-21, pp M1-M13.

Siau, K., Wand, Y., and Benbasat, I. 1997. "The relative importance of structural constraints and surface semantics in information modeling," *Information Systems* (22:2-3), April-May, pp 155-170.

Siau, K., and Rossi, M. 1998. "Evaluation of Information Modelling Methods - A Review," HICS (5) pp 314-322.

Siau, K., and Tan, X. 2005. "Evaluation Criteria for Information Systems Development Methodologies," *Communications of the Association for Information Systems* (16), pp 860-876.

Sommerville, I., and Kotonya, G. 1998. *Requirements Engineering: Processes and Techniques*. New York, NY: John Wiley & Sons, Inc.

Truex, D., Baskerville, R., and Travis, J. 2000. "Amethodical Systems Development: The Deferred Meaning of Systems Development Methods," *Accounting, Management, and Information Technologies* (10), pp 53-79.

Vidgen, R. 2002. "Constructing a web information system development methodology," *Information Systems Journal* (12), pp 247-261.

White, K. B., and Leifer, R. 1986. "Information Systems Development Success: Perspectives from Project Team Participants," *MIS Quarterly* (10:3), September, pp 214-223.

Yin, R. K. 1994. *Case study research: design and methods (2nd ed.)*. Sage Publications.

Appendix B: Questionnaires in the course integration case

The following appendixes contain copies of the questionnaires used in the course integration case. Please note that the room provided for filling in the answers are removed to shorten the questionnaires.

Q 1 Longitudinal questionnaire for students
Q 2 Summative questionnaire for student groups
Q 3 Summative questionnaire for teaching assistants
Q 4 Summative questionnaire for company representatives

Gruppenr:.....

Spørreskjema til styringsgruppemøtene

Hensikten med dette spørreskjemaet er:

- å gi hver student anledning til å reflektere over det som skjer i og rundt kurset og prosjektarbeidet
- å gi de kursansvarlige et bedre grunnlag for det videre arbeidet med prosjektbasert læring.

Du skal basere svarene på det som har skjedd siden siste milepæl. Som milepæler regnes oppgavegodkjenning og styringsgruppemøter. Begrepet "siste periode" er i skjemaet nedenfor brukt som betegnelse på dette tidsrommet. Svarene dine skal altså utelukkende være basert på det som har skjedd i denne perioden.

Dersom du har kommentarer til spørsmålene eller har andre kommentarer til siste periode som skjemaet ikke gir deg anledning til a å gi uttrykk for, ser vi gjerne at du gir slike kommentarer der du finner ledig plass.

Del 1:

Nedenfor finner du 12 ulike utsagn knyttet til prosjektarbeidet. Du skal angi i hvilken grad du er enig/uenig i disse utsagnene basert på siste periode.

	helt enig	delvis enig	delvis uenig	helt uenig
1. Jeg tror at jeg har lært mer gjennom prosjektarbeidet enn jeg ville ha gjort på tilsvarende tid i et mer Tradisjonelt kurs.	☐	☐	☐	☐
2. En del av det jeg har lært, ville jeg neppe ha lært i et mer tradisjonelt kurs	☐	☐	☐	☐
3. Prosjektarbeidet har ført til at jeg nå har fatt et noe annet syn på systemutvikling	☐	☐	☐	☐
4. Prosjektarbeidet har girt meg god anledning til å prøve ut det jeg har lært i andre kurs.	☐	☐	☐	☐
5. Oppdragsgiver har vært meget positiv/gitt meget gode bidrag til prosjektarbeidet.	☐	☐	☐	☐
6. Det har vært let å få kontakt med oppdragsgiver eller oppdragsgivers representanter, og de har start sett vært villige til å avse nødvendig tid til oss	☐	☐	☐	☐
7. Relativt stor frihet og ansvar i arbeidet med prosjektet, gjør at vi lærer mer av prosjektarbeidet.	☐	☐	☐	☐

8. Veiledningen fra hjelpelærere har, i den grad vi har benyttet slik veiledning i siste periode, vært meget nyttig for prosjektarbeidet	☐	☐	☐	☐
9. Veiledningen fra de kursansvarlige har, i den grad vi har benyttet slik veiledning i siste periode, vært meget nyttig for prosjektarbeidet	☐	☐	☐	☐
10. De kursansvarlige burde har vært mer bestemte i sin veiledning og gitt mer konkrete anvisninger på hvordan problemene møtes.	☐	☐	☐	☐
11. Prosjektbaserte.	☐	☐	☐	☐
12. Samarbeidet i prosjektgruppa har fungert meget bra.	☐	☐	☐	☐
13. Det har vært vanskelig å måtte forholde seg til både oppdragsgiver og kursansvarlige/hjelpelærere samtidig.	☐	☐	☐	☐

Del 2:

1. Hva synes du har vært mest positivt i arbeidet med prosjektet i siste periode?

Hva tror du er de viktigste årsakene til dette?

2. Hva synes du har vært vanskeligst/mest problematisk i arbeidet med prosjektet i siste periode?

Hva tror du er de viktigste årsakene til disse vanskelighetene/problemene?

3. Hva har du først og fremst lært/erfart gjennom arbeidet med prosjekt i siste periode?

4. Hvilke emner eller kurs fra studiet ditt har du hatt størst nytte av i arbeidet med prosjektet i siste periode?

5. Hvilke av følgende emner/omrader føler du at du har lært mest om gjennom arbeidet med prosjektet i siste periode? Ranger minst de 3 viktigste med henholdsvis 1, 2 og 3.
(Dersom du ønsker det, ser vi gjerne at du rangere flere på tilsvarende måte.):

☐ *konfliktløsning*
☐ *dokumentasjon og rapportskriving*
☐ *prosjektplanlegging og -administrasjon*
☐ *møteplanlegging og møteteknikk*
☐ *problemløsning*
☐ *systemutviklingsmetoder og -teknikker*
☐ *begrepsapparat og sammenhenger innen systemutvikling*
☐ *programmering gruppedynamikk*
☐ *mellommenneskelig kommunikasjon*
☐ *samarbeid*
☐ *databaser*
☐ *datamodellering*
☐ *brukergrensesnittdesign*
☐ *webdesign*
☐ *annet, spesifiser*

6. Burde noe i tilknytning til arbeidet med prosjektet i siste periode etter din mening ha vært gjort annerledes ut fra det du nå vet?

☐ Ja ☐ Nei

a) Hvis ja, hva burde du selv eventuelt ha gjort annerledes?

b) Hvis ja, hva burde prosjektgruppa eventuelt ha gjort annerledes?

c) Hvis ja, hva burde andre involverte eventuelt ha gjort annerledes (angi både hvem og hva?)

7. Hvordan føler du at du, prosjektgruppa og eventuelt andre involverte har klart å gjennomføre planer om endringer i organisering, arbeidsmåte o.l. som dere hadde ved forrige milepæl?

8. Hvilke forkunnskaper, ferdigheter og/eller erfaringer har du spesielt savnet i siste periode? Grunngi svaret.

Del 3:

Prosjektarbeidet i IS 2000 er organisert rundt sentrale element som gruppebasert arbeid, oppgavegodkjenning, veiledning, styringsgruppemøter o.l. Hvilken betydning mener du at de ulike organisatoriske elementene har hatt for prosjekt og læring i siste periode?

Ranger elementenes betydning (fra 1 til 1 0) i hver av kolonnene i tabellen nedenfor.

Organisasjonselement	Betydning for		
	forventet prosjekt-resultat	prosjekt-arbeidet i perioden	egen læring i perioden
ansvar for å skaffe oppgave			
gruppebasert arbeid			
gruppekontrakt			
oppgavegodkjenning			
kontrakt med oppdragsgiver			
styringsgruppemøte			
veiledning fra hjelpelærere			
veiledning fra kursansvarlige			
samarbeid med oppdragsgiver			
Forelesninger o.l.			

Gruppenummer:……

Evalueringsskjema

Evalueringsskjemaet består av to hoveddeler. Den ene delen tar for seg den aktuelle kursgjennomføringen, mens den andre har som formål å evaluere den spesielle undervisningsformen i kurset spesielt.

Dersom noen av gruppene ønsker å gi tilbakemeldinger som ikke dekkes av evalueringsskjemaet, kan disse gis der det er avsatt plass til det eller som vedlegg til skjemaet.

Del 1: Evaluering av den aktuelle kursgjennomføringen

I denne delen er spørsmålene knyttet til den konkrete gjennomføringen av IS-200 våren 1999.

1.1. Gruppeetablering og samarbeid i gruppa

1.1.1 Hvilken modell ble brukt for å etablere prosjektgruppa?
(*Ranger de aktuelle alternativene fra 1 og oppover dersom flere modeller lå til grunn for gruppeetableringen.)*

□ Valgte stort sett samme gruppesammensetning som ti tidligere kurs.

□ Gruppa ble satt sammen av studenter som kjente hverandre fra før, men som i liten grad hadde jobbet sammen i gruppe tidligere.

□ Gruppesammensetningen ble bestemt av den kompetanse som vi trodde det ville være nyttig å ha i gruppa.

□ Det ble lagt vekt på å finne gruppemedlemmer med omtrent samme ambisjonsnivå.

□ Gruppe ble bygd opp rundt en eller to studenter med en oppgaveide.

□Gruppeetableringen var relativt tilfeldig, måtte bare få dannet ei gruppe.

□ andre modeller (beskriv kort):

1.1.2 I hvilken grad diskuterte prosjektgruppa i startfasen hvordan samarbeidet i gruppa skulle foregå?

□ Grundig diskusjon □ noe diskusjon □ nokså begrenset diskusjon □ ikke i det hele tatt

1.1.3 I hvilken grad ble det fastsatt konkrete regler for samarbeid i gruppa?

□ I stor grad □ til en viss grad □ i liten grad □ ikke i det hele tatt

Når ble eventuelt slike regler fastsatt?
□ innledningsvis
□ underveis
□ både innledningsvis og underveis

1.1.4 Dersom det ble fastsatt regler for samarbeidet, hvilke sider ved samarbeidet gjaldt disse for?

1.1.5 Hvordan fungerte samarbeidet i gruppa?

□ Meget godt □ ganske bra □ middels □ nokså dårlig □ svært dårlig

1.1.6 I hvilken grad bar det overenstemmelse mellom måten samarbeidet fungerte på og de forventningene gruppa innledningsvis hadde til hvordan dette kom til å fungere?

□ I stor grad □ til en viss grad □ i liten grad □ ikke i det hele tatt

1.1.7 I hvilken grad ble arbeidsoppgavene og ansvar konkret fordelt mellom gruppemedlemmene?

□ I stor grad □ til en viss grad □ i liten grad □ ikke i det hele tatt

Når ble denne arbeidsfordelingen gjennomført?
- □ primært tidlig i prosjektet
- □ primært etter behov underveis i prosjektet
- □ både tidlig og underveis
- □ Annet (spesifiser)

1.1.8 Hvilke kriterier ble lagt til grunn for denne arbeidsfordelingen i prosjektgruppa?

(Dersom flere kriterier ble lagt til grunn ønskes det er rangering fra 1 og oppover)

- □ ønsker fra de enkelte gruppemedlemmene
- □ det enkelte gruppemedlems kompetanse
- □ ønsket om en rettferdig fordeling av arbeidsbelastning
- □ Annet (spesifiser)

1.1.9 Andre kommentarer til gruppeetablering og samarbeid i gruppa?

1.2: Valg av prosjektoppgave

1.2.1 Hvordan ble et knyttet kontakt med oppdragsgiver?

(D*ersom flere alternativ framgangsmåter ble brukt, ønskes det en rangering etter betydning.)*

- □ via familie, venner eller kjente
- □ gjennom arbeidsforhold for en eller flere av studentene i gruppa
- □ ved direkte forespørsel til oppdragsgiver
- □ via andre som hadde kjennskap til en oppgavemulighet
- □ Annet (spesifiser)

1.2.2 I hvilken grad ble det ved valget av oppgave lagt vekt på at oppgaven skulle være faglig interessant og relevant?

□ I stor grad □ til en viss grad □ i liten grad □ ikke i det hele tatt

1.2.3. Hvor mange oppgavealternativ ble vurdert?

□ En □ to □ tre □ mer enn tre

1.2.4. I hvilken grad bar kravet om prosjektgruppens arbeid med godkjenning av oppgaven nyttig for det etterfølgende prosjektarbeidet?

□ I stor grad □ til en viss grad □ i liten grad □ ikke i det hele tatt

1.2.5. I hvilken grad bar det overenstemmelse mellom de forventninger prosjektgruppa hadde til prosjektet og slik prosjektet ble?

□ I stor grad □ til en viss grad □ i liten grad □ ikke i det hele tatt

1.2.6. Laget prosjektgruppa innledningsvis et overslag over forventet arbeidsmengde i prosjektet?

□ ja □ delvis □ nei

I hvilken grad stemte dette overslaget overens med den virkelige arbeidsmengden i prosjektet?

□ I stor grad □ til en viss grad □ i liten grad □ ikke i det hele tatt

1.2.7. Andre kommentarer til valg av prosjektoppgave?

1.3. Forholdet til oppdragsgiver

Med oppdragsgiver menes her både oppdragsgiver selv og aktuelle representanter for oppdragsgiver.

1.3.1. Hvordan følte prosjektgruppa at oppdragsgivers interesse for prosjektet var?

□ Meget stor □ stor □ middels □ liten □ svært liten

1.3.2. I hvilken grad var oppdragsgiver tilgjengelig mår prosjektgruppa hadde behov for kontakt?

□ Stort sett alltid □ vanligvis □ noen ganger □ sjelden

1.3.3. I hvilken grad skapte forskjellig bakgrunn for oppdragsgiver og prosjektgruppe problemer for samarbeidet?

□ I stor grad □ til en viss grad □ i liten grad □ ikke i det hele tatt

1.3.4. Hvordan fungerte samarbeidet med oppdragsgiver ut fra en samlet vurdering?

□ Meget bra □ bra □ mindre bra □ dårlig

1.3.5. Hva var eventuelt positivt i forholdet til oppdragsgiver?

1.3.6. Hva var eventuelt negativt i forhold til oppdragsgiver?

1.3.7. Andre kommentarer om forholdet til oppdragsgiver:

1.4. Forholdet til hjelpelærerne

1.4.1. I hvilken grad var hjelpelærerne tilgjengelige?

□ I stor grad □ i rimelig grad □ i liten grad □ nesten aldri

1.4.2. I hvilken grad følte prosjektgruppa at den fikk den hjelpen den hadde behov for fra hjelpelærerne?

□ I stor grad □ i rimelig grad □ i liten grad □ nesten aldri

1.4.3. Var det spesielle områder hvor prosjektgruppa hadde behov for hjelp, men hvor hjelpelærerne ikke var i stand til å hjelpe?

1.4.4. Hvordan fungerte samarbeidet med hjelpelærerne ut fra en samlet vurdering?

□ Meget bra □ bra □ mindre bra □ dårlig

1.4.5. Hva var spesielt positivt i forholdet til hjelpelærerne?

1.4.6. Hva var eventuelt spesielt negativt i forholdet til hjelpelærerne?

1.4.7. Andre kommentarer om forholdet til hjelpelærerne:

1.5 Forholdet til de kursansvarlige

1.5.1. I hvilken grad var de kursansvarlige tilgjengelige når det var behov for kontakt?

□ I stor grad □ i rimelig grad □ i liten grad □ nesten aldri

1.5.2. I hvilken grad gikk prosjektgruppa den hjelp/veiledning som den forventet fra de kursansvarlige?

□ I stor grad □ i rimelig grad □ i liten grad □ nesten aldri

1.5.3. Hva var de viktigste grunnene til at det ble tatt kontakt med de kursansvarlige?

□ for å få bedre karakter på prosjektet
□ for å begrense eget ansvar
□ for å redusere usikkerhet
□ for å få et bedre prosjektresultat
□ Annet (spesifiser)

1.5.4. I hvilken grad var informasjon og veiledning som ble gitt av de kursansvarlige av verdi for prosjektet?

□ I stor grad □ i rimelig grad □ i liten grad □ ikke i de hele tatt

1.5.5. En viktig side ved prosjektorienterte kurs er ansvars, og myndighetsfordelingen mellom prosjektgruppe og de kursansvarlige. Hvordan synes prosjektgruppa at denne fordelingen var i det aktuelle prosjektet?

□ de kursansvarlige blandet ser i alt for liten grad inn i prosjektet
□ de kursansvarlige blandet ser i alt for stor grad inn i prosjektet
□ fordelingen var passe
□ de kursansvarlige kunne vært mer aktive i vårt prosjekt
□ de kursansvarlige kunne vært vesentlig mer aktive i vårt prosjekt

1.5.6. Hvordan fungerte samarbeidet med de kursansvarlige ut fra en samlet vurdering?

□ Meget bra □ bra □ mindre bra □ dårlig

1.5.7. Hva var eventuelt spesielt positivt i forholdet til de kursansvarlige?

1.5.8. Hva var eventuelt spesielt negativt i forholdet til de kursansvarlige?

1.5.9. Andre kommentarer om forholdet til de kursansvarlige:

1.6. Undervisning og pensum

1.6.1. Hvordan var størrelsen på det samlede undervisningsomfanget?

□ altfor stort □ for stort □ passe □ for lite □ alt for lite

1.6.2. Hva mener prosjektgruppa om følgende sider ved pensumlitteraturen?

a) omfang

□ altfor stort □ for stort □ passe □ for lite □ alt for lite

b) relevans

□ Meget relevant □ rimelig relevant □ delvis relevant □ lite relevant

c) vanskelighetsgrad

□ For vanskelig □ passe □ for lett

d) annet (beskriv):

1.6.4. Hva var det samlede inntrykket av undervisningen i kurset?

□ Meget bra □ over middels □ middels □ under middels □ dårlig

1.6.5. Hva var eventuelt spesielt positivt ved undervisningen og pensum?

1.6.6. Hva var eventuelt spesielt negativt ved undervisningen og pensum?

1.6.7. Andre kommentarer til undervisning og pensum:

1.7. Helhetsvurdering

1.7.1. Hva er prosjektgruppas helhetsinntrykk av kurset?

□ Meget bra □ over middels □ middels □ under middels □ dårlig

1.7.2. Hvor relevant er dette kurset i forhold til studiet?

□ Meget relevant □ nokså relevant □ litt relevant □ lite relevant

1.7.3. Hvordan var det faglige utbyttet av kurset?

□ Meget stort □ stort □ middels □ lite □ svært lite

1.7.4. Hvordan var arbeidsbyrden i kurset?

a) i forhold til antatt vekttall

□ altfor stor □ litt for stor □ passe □ litt for liten □ alt for liten

b) i forhold til det faglige utbyttet av kurset

□ altfor stor □ litt for stor □ passe □ litt for liten □ alt for liten

1.7.5. Hva var det først og fremst som bestemte arbeidsbyrden?

□ uklare og stadig nye/endrede krav fra oppdragsgiver
□ utilstrekkelig avgrensning av prosjektet
□ manglende forkunnskaper
□ liten erfaring med prosjektarbeid
□ ikke tilstrekkelig effektivitet i prosjektarbeidet
□ Annet (spesifiser):

1.7.6. I hvilken grad føler prosjektgruppa at arbeidet i gruppa var effektivt?

□ i stor grad □ til en viss grad □ i liten grad

1.7.7. Var det noe eller noen som skapte spesielle problemer i forbindelse med prosjektgjennomføringen?

1.7.8. I hvilken grad stilte kurset kvar til forkunnskaper som prosjektgruppa i utgangspunktet ikke hadde?

□ I stor grad □ til en viss grad □ i liten grad □ ikke i det hele tatt

Hvilke typer forkunnskaper savnet prosjektgruppa spesielt?

1.7.9. Hva var prosjektgruppas oppfatning av størrelsen på de tilgjengelige ressursene i dette kurset?

□ Alt for mye □ litt for mye □ passe □ litt for lite □ alt for lite

1.7.10. I hvilken grad utnyttet prosjektgruppa de tilgjengelige ressursene?

□ I stor grad □ i rimelig grad □ i liten grad □ ikke i det hele tatt

Hvorfor ble de tilgjengelige ressursene eventuelt ikke utnyttet?

1.7.11. Savnet prosjektgruppa spesielle ressurser som ikke var tilstede?

□ ja, i tilfelle hvilke?
□ nei

1.7.12. I hvilken grad følte prosjektgruppa et forventningspress i forbindelse med prosjektet?

□ I stor grad □ i rimelig grad □ i liten grad □ ikke i det hele tatt

Hvem var eventuelt det primære opphavet til dette forventningspresset?

□ dere selv
□ oppdragsgiver
□ de kursansvarlige
□ hjelpelærere
□ Andre (spesifiser)

1.7.13. Andre kommentarer til helhetsvurderingen?

Del 2: Evaluering av undervisningsformen

I denne delen er vi interessert i prosjektgruppas meninger om prosjektorientert undervisning som undervisningsform. Spørsmålene er derfor relativt prinsipielle og ikke direkte knyttet til den aktuelle kursgjennomføringen. Her der det ønskelig at prosjektgruppa forsøker å generalisere lik at svarene ikke blir for direkte preget av den aktuelle kursgjennomføringen.

2.1. Hvordan er prosjektgruppas holdning til prosjektorientert undervisningen/prosjektorienterte kurs?

□ Meget positiv □ positiv □ nøytral □ negativ □ meget negativ

2.2. I hvilken grad bør prosjektorientert undervisning benyttes i studier som dette?

□ I stor grad □ il en viss grad □ i begrenset grad □ ikke i det hele tatt

2.3. Bør det være flere prosjektorienterte kurs enn det som er tilfelle i dag?

□ ja □ kanskje □ nei □ vet ikke

2.4. Hva er positivt med denne undervisningsformen sammenlignet med tradisjonell undervisning?

2.5. Hva er negativt med denne undervisningsformen sammenlignet med tradisjonell undervisning?

2.6. Prosjektene i et prosjektorientert kurs kan komme fra reelle oppdragsgivere (som i IS 200) eller være skoledefinerte prosjekt. Hva slags prosjekt foretrekker prosjektgruppa og hvorfor?

2.7. I hvilken grad har studiet forberedt prosjektgruppas medlemmer på de utfordringene prosjektarbeidet bød på?

□ I stor grad □ i rimelig grad □ i liten grad □ i svært liten grad

2.8. Bør studentene selv ha ansvaret for gruppeetablering og for å skaffe oppgave?

□ ja, hvorfor:
□ nei, hvorfor ikke
□ vet ikke

2.9. Bør prosjektoppgavene innledningsvis godkjennes av de kursansvarlige?

□ ja, og kort hvorfor:
□ nei, og kort hvorfor ikke

□ vet ikke

2.10. Bør det være møter met ei styringsgruppe (eller tilsvarende møter om framdriften) underveis i prosjektet?

□ ja, og kort hvorfor:

Hvor mange møter bør det være?

□ nei, og kort hvorfor ikke:
□ vet ikke

2.11. Bør prosjektgruppene ha en egen prosjektleder?

□ ja, og kort hvorfor:
□ nei, og kort hvorfor ikke:
□ vet ikke

2.12. Bør prosjektlederen være fast, eller bør prosjektlederansvaret rullere mellom gruppemedlemmene gjennom prosjektet?

□ fast
□ rullere
□ Annet (spesifiser):

2.13. Andre kommentarer til undervisningsformen:

3. Andre kommentarer fra prosjektgruppa:

4. Individuelle kommentarer:

Til hjelpelærerne i IS 200 våren 1999

Vi gjennomfører denne gang en relativt omfattende evaluering av IS 200: Hensikten er for det første å ha en tradisjonell evaluering av denne kursgjennomføringen. Denne måten å gjennomføre kurset på var også til dels en prøveordning, og den skal derfor evalueres av instituttet. Selv er vi interessert i prosjektorientert undervisning som undervisningsform. Vi ønsker derfor å se nærmere på denne formen for undervisning med utgangspunkt i IS 200.

Som vanlig har vi bedt studentene evaluere kurset. Hver prosjektgruppe har fått et ganske omfattende evalueringsskjema. For å få et best mulig grunnlag for det videre arbeidet, ønsker vi også tilbakemelding og synspunkter fra de andre involverte. Vi har derfor utarbeidet egne evaluerings-/spørreskjema både for hjelpelærere og oppdragsgivere.

Nedenfor finner du et evalueringsskjema beregnet på hjelpelærere. Vi har forsøkt å få med det vi mener det er viktig a spørre hjelpelærerne om. Det kan godt tenkes at vi har oversett noe. I sa fall ber vi om at dere tilføyer kommentarer og synspunkter til skjemaet.

Skjemaet omfatter både spørsmål som er knyttet direkte til den aktuelle kursgjennomføringen og spørsmål av noe mer generell karakter.

De kommentarer og synspunkter vi mottar, vil ikke bli brukt i forbindelse med sensur av prosjektoppgavene.

Med et så begrenset antall hjelpelærere kan det vanskelig unngås at vi stort sett vet hvem av dere som har svart hva. I all presentasjon av svarene eller annen bruk av dem vil imidlertid svarene bli anonymisert i den grad det er praktisk mulig. For andre vii det derfor være svært vanskelig å henføre enkeltsvar direkte til en av dere.

Evalueringsskjemaene vil bli oppbevart på en betryggende måte inntil vi er ferdige med behandlingen av dem. Deretter vil de bli destruert.

Vi håper at du har anledning til å svare på spørsmålene i evalueringsskjemaet. Av hensyn til den videre behandlingen, ville det vært fint om vi kunne ha fått skjemaet tilbake innen 5. mai.

Med prosjektorientert hilsen

Hans Olav og Tore

Evalueringsskjema

Del 1: Studentegenskaper o.l.

1.1. Hvordan føler du at studentene var forberedt til å takle de utfordringene som prosjektene gav?

□ I stor grad □ i rimelig grad □ i noen grad □ i liten aldri

1.2. Var det noe som du mener de var spesielt godt forberedt til?

1.3. Var det noe som du mener de var spesielt dårlig forberedt til?

1.4. Var det forkunnskaper som studentene hadde behov for, men som få av dem hadde?

□ ja, hvilke:
□ nei
□ vet ikke

1.5. Hvordan feler du at studentene totalt sett var motivert for prosjektene?

□ Meget godt □ godt □ middels □ dårlig □ meget dårlig

1.6. Hvor føler du at forventningspresset for prosjektgruppene primært kom fra?
(Ranger aktuelle alternativ fra 1 og oppover.)

□ fra prosjektgruppa selv
□ fra oppdragsgiver
□ fra de kursansvarlige
□ fra hjelpelærerne
□ fra andre (spesifiser):

1.7. I hvilken grad mener du at prosjektgruppene klarte å arbeide effektivt med prosjektene?

□ Meget stor □ stor □ middels □ liten □ meget liten

1.8. Var det etter din mening noe spesielt som eventuelt hindret prosjektgruppene i å arbeide effektivt?

Del 2: Gruppeetablering, valg av oppgave

2.1. Studentene hadde selv ansvaret for a å etablere prosjektgruppene. Bør det etter din mening være slik?

□ ja, hvorfor

□ nei, h v o r for ikke:

□ vet ikke

Andre synspunkter på gruppeetableringen:

2.2. Studentene hadde selv ansvaret for a å finne oppdragsgiver og oppgave. Bør det etter din mening være slik:?

□ ja, hvorfor:
□ nei, hvorfor ikke:
□ vet ikke

Andre synspunkter på det a å finne oppdragsgiver og oppgave:

Del 3: Veiledningsarbeidet

3.1. I hvilken grad føler du at det var overensstemmelse mellom prosjekt gruppenes og din oppfatning av hva veiledningen skulle være?

□ I stor grad □ i rimelig grad □ til en viss grad □ i liten aldri

Andre kommentarer:

3.2. I hvilken grad klarte de kursansvarlige å formidle sin oppfatning av hva veiledning burde være?

□ I stor grad □ til en viss grad □ i liten aldri □ ikke i det hele tatt

Kommentar:

3.3. I hvilken grad klarte studentene a å gi deg en s k i k k e l i g problembeskrivelse og nødvendig bakgrunnsinformasjon når de ba om veiledning?

□ I stor grad □ i rimelig grad □ til en viss grad □ i liten aldri

Hva burde de eventuelt i denne sammenheng ha vært flinkere til?

3.4 I hvilken grad føler du at du måtte hjelpe prosjektgruppene med å finne ut av hva problemet egentlig var?

□ I stor grad □ i rimelig grad □ til en viss grad □ i liten aldri

3.5. I hvilken grad føler du selv at du var tilgjengelig for prosjektgruppene når de hadde behov for veiledning?

□ I stor grad □ i rimelig grad □ til en viss grad □ i liten aldri

3.6. I hvilken grad foler du at det er behov for ansikt-til-ansikt-kontakt mellom hjelpelærer og prosjektgruppe?

□ I stor grad □ til en viss grad □ i liten aldri

3.7. I hvilken grad gikk samme problem igjen hos ulike grupper?

□ I stor grad □ til en viss grad □ i liten aldri

Hvilke problem var det eventuelt som gikk igjen?

3.8. Var det noe som skapte spesielle problemer i veiledningsarbeidet?

3.9 Føler du at veiledningsarbeidet stilte krav til kompetanse som du ikke hadde?

□ ja, hvilke områder:

3.10. Andre kommentarer om veiledningsarbeidet:

Del 4: Informasjon og koordinering

4.1. Var den informasjonen som initielt ble gitt om din hjelpelærerstilling tilstrekkelig?

□ ja
□ nei, Hva savnet du:
□ vet ikke

4.2. Føler du at du som hjelpelærer innledningsvis å ha fatt mer informasjon om de prosjektene som du skulle veilede?

□ ja
□ ne1
□ vet ikke

Hvordan burde denne informasjonen ha vært gitt?
(Ranger fra 1 og oppover dersom du velger flere alternativ.)

□ ved deltakelse på styringsgruppemøtene for "dine" prosjektgrupper
□ ved å få kopier av framdriftsrapporter o.l. fra styringsgruppemøtene
□ ved regelmessig informasjon fra de kursansvarlige

Spesifiser:

4.3. Burde det ha vært et eller flere moter med hjelpelærere og kursansvarlige i løpet av prosjektperioden for å utveksle informasjon og for å drøfte felles problem?

□ ja Hvor mange:
□ nei
□ vet ikke

4.4. Andre kommentarer om informasjon og koordinering:

Del 5: Ressurstilgangen

5.1. Hvordan foler du at den samlede hjelpelærerressursen som var tildelt ditt ansvarsområde var i forhold til behovet?

□ Alt for stor □ noe for stor □ passe □ litt for liten □ alt for liten

5.2. Hjelpelærerressursene ble i utgangspunktet fordelt på de ulike gruppene. Var denne fordelingen til nyttig i arbeidet ditt?

□ I stor grad □ til en viss grad □ i liten aldri □ ikke i det hele tatt

5.3. Andre kommentarer til ressurstilgang og fordeling:

Del 6: Kvaliteten på prosjektarbeidet og prosjektresultatet

6.1. Hva er din oppfatning av den gjennomsnittlige kvaliteten på prosjektarbeid og prosjektresultat?

□ Meget høy □ middels □ lav □ meget lav

Kan du si noe mer om hvordan kvalitetsfordelingen var?

6.2. Er det spesielle omrader når det gjelder kvalitet som etter din mening burde ha vært bedre?

Del 7: Andre kommentarer og synspunkter, forslag til forbedringer:

Nummer: .....

1. Evalueringsskjema Oppdragsgivers representant

1.1. Valg av prosjektoppgave

1.1.1. Hvordan ble det knyttet kontakt mellom oppdragsgiver og prosjektgruppe?

(Dersom flere, alternative fremgangsmåter ble brukt, ønskes det en rangering etter betydning.)

☐ via familie, venner eller kjente

☐ gjennom arbeidsforhold for en eller flere av studentene i gruppa

☐ ved direkte forespørsel til oppdragsgiver

☐ via andre som hadde kjennskap til en oppgavemulighet

☐ Annet (spesifiser): .....

.....

1.1.2. Prosjektideen skulle godkjennes av prosjektansvarlige ved Høgskolen prosjektet formelt ble satt i gang. I hvilken grad mener du det er ønskelig?

☐ i stor grad ☐ til en viss grad ☐ i liten grad ☐ ikke i det hele tatt

1.1.3. I hvilken grad mener du kravet om og prosjektgruppens arbeid med godkjenning oppgaven var nyttig for det etterfølgende

☐ i stor grad ☐ til en viss grad ☐ i liten grad ☐ ikke i det hele tatt

Eventuelle kommentarer om prosjektgodkjenningen:

1.1.4. Fikk du som oppdragsgiver innledningsvis nok informasjon om prosjektoppgaven til å kunne ta en begrunnet avgjørelse om deltakelse i prosjektet?

☐ i stor grad ☐ til en viss grad ☐ i liten grad ☐ ikke i det hele tatt

1.1.5. Var det noe annet du som oppdragsgiver savnet eller ønsket kontakt?

1.1.6. I hvilken grad var det overensstemmelse mellom de forventninger du innledningsvis hadde til prosjektet og slik prosjektet virkelig ble?

☐ i stor grad ☐ til en viss grad ☐ i liten grad ☐ ikke i det hele tatt

1.1.7. Hvordan mener du forventningene dine ble skapt?
(Prioriter alternativene dersom du mener at flere alternativer har innvirket på deg)

☐ Innflytelse fra prosjektgruppa

☐ Dine egne ønsker/tanker/ideer

☐ Tidligere erfaringer med prosjektarbeid

☐ Annet (spesifiser) .....

1.1.8. Presenterte prosjektgruppa innledningsvis et overslag over forventet arbeidsmengde for deg iprosjektet?

☐ ja I hvilken grad opplevde du at dette overslaget stemte overens med din virkelige arbeidsmengde i prosjektet?

☐ i stor grad ☐ til en viss grad ☐ i liten grad ☐ ikke i det hele tatt

☐ delvis

☐ nei

1.1.9. Andre kommentarer angående valg av prosjektoppgave:

1.2. Forholdet til prosjektgruppa

Med prosjektgruppa menes her både prosjektgruppa som enhet og de enkelte studentene.

1.2.1. Hvordan følte du at prosjektgruppas interesse for prosjektet var?

☐ Meget stor ☐ stor ☐ middels ☐ liten ☐ svært liten

1.2.2. I hvilken grad mener du at du eller andre normalt var tilgjengelig når prosjektgruppa hadde behov for kontakt med oppdragsgiver?

☐ Stort sett alltid ☐ i rimelig grad ☐ i begrenset grad ☐ sjelden

1.2.3 Hvilken form hadde kontakten?
(Prioriter fra 1 og oppover etter hyppighet)

☐ avtalte m0ter

☐ ikke avtalte møter

☐ e-post

☐ telefon

☐ telefax

☐ annet (spesifiser).....

1.2.4. I hvilken grad opplevde du at eventuell forskjell i bakgrunn mellom deg og prosjektgruppa skapte problemer for samarbeidet?

☐ i stor grad ☐ til en viss grad ☐ i liten grad ☐ ikke i det hele tatt

1.2.5. Hvordan følte du studentene var forberedt på denne type prosjekt?

☐ i stor grad ☐ til en viss grad ☐ i liten grad ☐ ikke i det hele tatt

Angi om det er omrader der du opplevde de var godt og/eller dårlig forberedt, kunnskaper du savnet o.l.:

1.2.6. I hvilken grad tror du at prosjektgruppa opplevde forventningspress fra oppdragsgiver?

☐ i stor grad ☐ til en viss grad ☐ i liten grad ☐ ikke i det hele tatt

1.2.7. Hvordan synes du samarbeidet med prosjektgruppa fungerte ut fra en samlet vurdering?

☐ meget bra ☐ bra ☐ mindre bra ☐ dårlig

1.2.8. Kommentarer, forslag til forbedringer osv. i forhold til prosjektgruppe:

1.3. Styringsgruppemøtene/ansvars- og myndighetsfordeling

1.3.1 Synes du at møtene med de kursansvarlige rett før styringsgruppemøtene var nyttige?

☐ i stor grad ☐ til en viss grad ☐ i liten grad ☐ ikke i det hele tatt

Hva synes du eventuelt var spesielt nyttig ved dem?

1.3.2. I år er første gang styringsgruppemøter inngår som en del av prosjektet-gjennomføringen. Hvilken nytte mener du disse møtene har hatt for prosjektet?

☐ meget stor ☐ stor ☐ middels ☐ liten ☐ meget liten

Hvilken nytte synes du styringsgruppemøtene har innen de nedenstående områdene:

	Stor	middels	liten	ingen
Læringsforum	☐	☐	☐	☐
Ideutviklingsforum	☐	☐	☐	☐
Beslutningsforum	☐	☐	☐	☐
Diskusjonsforum	☐	☐	☐	☐
Annet (spesifiser)				
……	☐	☐	☐	☐
……	☐	☐	☐	☐
……	☐	☐	☐	☐
Nytte for studentene	☐	☐	☐	☐
Nytte for oppdragsgiver	☐	☐	☐	☐

1.3.3. Andre konm1entarer til styringsgruppemøtene, forslag til forbedringer

1.3.4. En viktig side ved prosjekter er ansvars- og myndighetsfordelingen mellom prosjektgruppe og oppdragsgiver.

I hvilken grad tok prosjektgruppa ansvar og initiativ:

☐ ☐ ☐ ☐ ☐
Alt for liten for liten passe for stor alt for stor

Hvordan opplevde du prosjektgruppas vilje og evne til å ta beslutninger:

☐ ☐ ☐ ☐ ☐
Alt for liten for liten passe for stor alt for stor

I hvilken grad tok prosjektgruppa beslutninger uten å rådføre seg med deg på omrader der du syntes beslutningene burde vært fattet i fellesskap eller av deg?

☐ ☐ ☐ ☐ ☐
i stor grad til en viss grad i liten grad for lite ikke i dett hele tatt

Tok du tilstrekkelig initiativ overfor prosjektgruppa?

☐ ☐ ☐ ☐ ☐
Alt for lite for lite passe for mye alt for mye

1.3.5. En viktig side ved prosjektorienterte kurs er ansvars- og myndighetsfordelingen mellom prosjektgruppe og de kursansvarlige. Hvordan synes du at denne fordelingen kom til syne?

De kursansvarliges innblanding/aktivitet var:

☐ ☐ ☐ ☐ ☐
Alt for liten for liten passe for stor alt for stor

De kursansvarliges innvirkning og setting av premisser for prosjektgruppa var:

☐	☐	☐	☐	☐
alt for liten	for liten	passe	for stor	alt for stor

1.3.6. Kommentarer i forhold til ansvars- og myndighetsfordelingen:

1.4. Applikasjonen

Med applikasjonen menes her det programvareprodukt som prosjektgruppa utviklet.

1.4.1. Installasjon og demonstrasjon av applikasjonen (per dato):

Er applikasjonen

Installert	☐ ja	☐ nei
Planlagt installert	☐ ja	☐ nei

Har prosjektgruppa demonstrert for oppdragsgiver

den endelige versjonen av applikasjonen	☐ ja	☐ nei
avtalt demonstrasjon av den endelige versjonen av applikasjonen	☐ ja	☐ nei

Har oppdragsgiver gjort seg kjent med applikasjonen på egenhånd ☐ ja ☐ nei

Andre kommentarer om installasjon av applikasjon:

1.4.2. Bruk av applikasjonen (per dato):
(Alternativene under kan sees som gjensidig utelukkende)

Applikasjonen

er tatt i bruk	☐ja	☐nei
er planlagt tatt i bruk	☐ ja	☐nei
vii muligens tas i bruk	☐ja	☐nei
vil være et grunnlag for videre utvikling av en applikasjon	☐ ja	☐nei

annet:……..

Dersom ingen av de ovenstående alternativer passer:

I hvilken grad har deltakelse i prosjektarbeidet vært nyttig for deg?

□	□	□	□
i stor grad	til en viss grad	i liten grad	ikke i det hele tatt

Hva har vært spesielt nyttig?

Andre kommentarer om bruk av applikasjon:

1.4.3 Eventuell videre kontakt med prosjektgruppa eller enkeltmedlemmer i gruppa:
Oppdragsgiver

har gjort avtale om videre kontakt	☐ ja	☐ nei
planlegger videre kontakt	☐ ja	☐ nei
ønsker videre kontakt	☐ ja	☐ nei

Kontakten gjelder:

hele prosjektgruppa	☐ ja	☐ nei
for videreutvikling	☐ ja	☐ nei
for drift og vedlikehold	☐ ja	☐ nei

1.4.4. Opplevde du at dine forventninger til det ferdig produkt forandret seg underveis?

☐	☐	☐	☐
i stor grad	til en viss grad	i liten grad	ikke i det hele tatt

Ble forventningene: ☐ større ☐ mindre

Hva tror du eventuelt var grunnen til denne forandring?
(rangér dersom du ønsker å bruke flere alternativer)

□ prosjektgruppa
□ egne ambisjoner
□ egen læring
□ så flere muligheter etterhvert
□ annet (spesifiser).....

1.4.5. Hva mener du først og fremst bestemte arbeidsbyrden for prosjektgruppa?
(Ranger fra 1 og oppover.)

□ utilstrekkelig avgrensning av prosjektet
□ manglende forkunnskaper
□ liten erfaring med prosjektarbeid
□ ikke tilstrekkelig effektivitet i prosjektarbeidet
□ uklare og stadig nye/endrede krav fra oppdragsgiver
□ krav/ønsker både fra kursansvarlige og oppdragsgiver
□ annet (spesifiser): .....

1.4.6. Andre kommentarer i forhold til applikasjonen?

1.5. Helhetsvurdering

1.5.1. Hva er din opplevelse av prosjektarbeid som undervisningsform for studenter som dette?

□ Meget god □ god □ middels □ dårlig □ meget dårlig

1.5.2. Hvilken verdi mener du erfaringer fra et slikt prosjekt er i forhold til en framtidig arbeidssituasjon for studentene?

□ Meget stor □ stor □ middels □ liten □ meget liten

1.5.3. Hvordan var din/virksomhetens arbeidsbyrde i prosjektet:

a) i forhold til resultat

□ Alt for stor □ litt for stor □ passe □ litt for liten □ alt for liten

b) i forhold til det du måtte ha lært

□ Alt for stor □ litt for stor □ passe □ litt for liten □ alt for liten

c) i forhold til din interesse for det problemområdet prosjektet omhandlet

□ Alt for stor □ litt for stor □ passe □ litt for liten □ alt for liten

d) annet

1.5.4. I hvilken grad synes du at arbeidet i prosjektgruppa var effektivt?

□ i stor grad □ til en viss grad □ i liten grad

1.5.5. Var det noe eller noen som du føler skapte spesielle problemer i forbindelse med prosjektgjennomføringen?

1.5.6. Savnet du ressurser som ikke var tilgjengelige i prosjektet og som kursansvarlige burde skaffet til veie?

□ ja, tilfelle hvilke:

□ nei
□ vet ikke

1.5.7. Kjente du et forventningspress i forbindelse med prosjektarbeidet?

□ stort □ rimelig □ lite

Hvilken innvirkning tror du et slikt press hadde på studentene?

Hvem mener du eventuelt var det primære opphavet til dette presset?
(Ranger fra 1 og oppover dersom du angir flere.)

□ deg/dere selv
□ prosjektgruppa
□ de kursansvarlige
□ hjelpelærerne
□ Andre (spesifiser): .....

1.5.8. Andre kommentarer til helhetsvurderingen av prosjektet:

1.5.9. Dersom din virksomhet har et egnet problem/prosjekt, vil du da vurdere å være med på et nytt prosjekt i framtiden?

□ ja □ nei

Hvordan ønsker du å oppnå kontakt med aktuelle prosjektgrupper?

□ Melde interesse via kursansvarlige
□ Bli oppsøkt av studenter
□ Kursansvarlige formidler en generell invitasjon til studentene
□ Annet:

□ nei
□ vet ikke

1.5.10. Ønsker du å få tilgang til resultatene av denne spørreundersøkelsen?

□ ja
□ nei

2. Andre kommentarer fra oppdragsgiver:

Takk for hjelpa og for at du tok deg tid til å fylle ut og returnere spørreskjemaet.

Appendix C: List of documents studied

The following documents were studied in the course integration case and the municipality case:

1. **The course integration case**

   The following documents from the following courses were studied:

   IS-2000 Project Work – Application Development

   The course description and course syllabus
   The feed-back forms from the steering committee meetings
   The project reports produced by the students

   IS-3000 Project – Application Development

   The course descriptions and the course syllabuses for courses:

   IS-3200 Project work and Quality Assurance

   The course descriptions and the course syllabuses for courses:

2. **The municipality case**

   The process-oriented requirement specification document made by the municipality

Appendix D: Interview guide

Interview guide for the government agency case and the municipality case

Interview guide (08.06.04)
(Based on Patton)

Goal:

To describe the relationship between actors' competence, methods and practice in the process of customising, implementing and using the new administrative system in Kristiansand kommune.

Expected outcome of the interview:

The interviewee's experience and thoughts about the project including direct and indirect comments on actors' competence, methods and practice and the relationship between these.

Method:

General interview in relation to one big project
Selecting interviews in relation to the analysis activities in the development process

Questions, areas to touch:

1. Describe the project from your point of view
   a. What is the project?
   b. Use
   c. Context
   d. Etc.
   e.

2. Describe your involvement in the project

3.
4. Planning activities
   a. Of analysis activities
      i. When was the planning carried out? How long time did it take?
      ii. What activities did you do in the planning process?
   b. Of design activities
   c. Of customisation activities
   d. Of implementation activities
   e. Of use
   f.
5. "Body":
   1. How did you staff the project?
      - Staff worked together before?
      - Changes in staffing during the project?
      - What did you want to obtain by the change?
      - Consequences of the change? How will you describe the result you got?

2. What considerations did you do about competencies when you staffed the project?
   - How would you describe the actual use of competencies/staff?
   - Changing the competence?
   - How do you value the changes?
3. How did you decide what methodologies to use?
   - What did you intend to use the methodologies for?
   - How did the use of methodologies contribute to the project, the Systems development?
4. The transfer from the analysis phase/activities to the design and following phases?
5. Any consequences in the later phases that can be related to the analysis phase/activities?
6. How do you value the project as a whole? (Successful, failure)
   - What do you base your evaluation on?
7. How did you obtain the competencies you used? Availability of competencies?

6. Generally, if you had to do things differently:
   How did you find that you needed to do thing differently from what you originally set out to act?
   1. Specifically regarding staffing?
   2. Specifically regarding methodologies?
   3. Specifically regarding competencies?

7. Follow up on Question 5 in execute the changes and evaluate the results

8. What, if any competencies, will you rate as the most important competencies contributing to the success or lack of competencies leading to the failure of the project?

9. What are your understanding/thoughts about competence? Contents/domain vs. process?

10. How do you view the relationship between methodologies and competencies?

11. Value decisions?

12. Reflections?

13. Personalia background etc.

Question:

What is the three most important “styringsparametre” for the analysis phase of the project?”