Learning Cycles, Organizational Back Talk, and the Persistence of Theories in Use: Lessons of Information Systems Development in a University Administration Context

Ari Heiskanen
Department of Information Processing Sciences, University of Oulu, Finland

Pauli Assinen
Information Systems Services, IT Department, University of Helsinki

Mailing address for correspondence

Ari Heiskanen
P.O.Box 3000
FIN-90014 UNIVERSITY OF OULU
FINLAND

Telephone +358-40-5056914

E-mail Ari.Heiskanen@Oulu.Fi

Learning Cycles, Organizational Back Talk, and the Persistence of Theories in Use: Lessons of Information Systems Development in a University Administration Context

Abstract

Learning experiences of twenty-eight administrative information system development projects in a large Finnish university are analysed over a time frame of eighteen years. A series of reporting development projects is chosen for more detailed inspection. Organisational learning is conceptualised to consist of cycles during which the organisational reactions towards development actions reveal the successfulness of each project. In a retrospective analysis the authors find three learning cycles out of the twelve-year long reporting development process. The authors define their approach as reflective information systems practice and relate it to action learning and action science.

1. Introduction

It is well known that information systems (IS) development is risky and a lot of IS projects have failed to deliver their promises: they have exceeded their budget, schedule, or both, or even worse, some systems never become operational. Organizational learning is related to IS development in two ways (Robey, Boudreau and Rose 2000): first, implementing an IS necessarily entails organizational learning, and second, information technology can be designed to be a part of the organizational memory that supports organizational learning. An intriguing feature is that information technology has the potential to both enable and disable (e.g. Gill 1995) organizational learning. In many cases actors are persistent in pursuing a failing action strategy for many years, thus showing poor capability to learn (Robey and Newman 1996; Lyytinen and Robey 1999).

In this article we report of our analysis of the histories of about thirty IS development projects in the administration of the University of Helsinki. The time frame of our study covers the years 1985 – 2001. Our aim is to learn what has been successful in this context and why, and why some projects failed. Only four projects were total failures, because the developed system was never used. Several projects have exceeded their original schedule and budget, and encountered a lot of problems.

Our study is related to the previous work of the first author as the chief information systems officer in the University of Helsinki (Heiskanen 1995; Heiskanen and Similä 1992; Heiskanen and Newman 1997; Heiskanen, Newman and Similä 2000). As the first author now has moved at least temporarily to an academic position in another university, this article can be seen also as an attempt to transfer the expertise of the first author to the second one who continues in the capacity of the chief information systems officer.

We will be using the notions developed by Schön (1983, 1987) and Argyris (e.g. Argyris, Putnam and Smith 1987, Argyris and Schön 1978) when we try to figure out what has been the essence of our experiences. Using the notions developed by Nonaka and his colleagues (Nonaka and Takeuchi 1995; Nonaka, Toyama and Konno 2000) we try to formulate our experiences in such a way that they make sense for a broader audience. We also relate our experiences to action learning and action science (Raelin 1997a).

We believe that our work will be interesting for people coming from practice and academia at least in two fields: adult education and information system studies, because we can present insiders' view of how the capability to manage IS development projects can be conceptualised and researched. Moreover, we present example of the mechanisms that have enabled or disabled organizational learning. Our aim is also to develop the capabilities of the information systems services unit of the University by creating means to retrieve efficiently past experience and renew this organizational knowledge in new IS projects. An organizational "memory" in the form of a project history and template archive is planned to be developed as one result of our work.

The article in its current form is aimed to be a discussion paper during the OKLC 2002 Conference. We seek feedback to our work especially from those participants that are familiar with action learning and action science. As both authors are from the IS field, it would be interesting to relate our experiences and interpretations to the views of other fields of research.

2. Our Research Approach

Our aim is to investigate the possibilities to put our direct experience from practice into a form that makes sense to both academic and practical audience (Heiskanen and Newman 1997). For this purpose we use the notions of Reflection-in-Action, adopted from Schön (1983, 1987), Raelin (1997b, 2001), and Seibert (1995). Our task is what Nonaka and Takeuchi (1995), and Nonaka, Toyama, and Konno (2000) call knowledge conversion from unarticulated practice to explicit knowledge. We will rely on simple methods, tracing the flow of events and decisions in an organisation. Our approach is not a rare one, because there exists other examples of how to reflect over development projects (e.g. Ayas and Zeniuc 2001). Indeed, reflection and reflective practice have experienced a boom since the seminal work of Schön (1983). For example, recently a journal called "Reflective Practice" was founded, and other journals give space to these themes, like a special issue of "Management Learning" in 2001.

Reflection is the practice of periodically stepping back to ponder self and others in one's immediate environment (Raelin 2001). The object of reflection may be in three areas. First, content reflection is about how a practical problem was solved. Second, process reflection examines the procedures and sequence of the events. Third, premise reflection goes to questioning the presuppositions attending to the problem. The timing of reflection may be anticipatory, contemporaneous, or retrospective. Originally, Schön (1983, p. 163) characterised the work of design as a reflective conversation with the situation where the practitioner functions as an agent and experient[1]. He coined this as reflection-in-action.

Schön described the structure of the reflective process as follows (1983, pp. 129 - 132). The practitioners approach the practice problem as a unique case. Through their transactions with the situations, they shape it and make themselves a part of it. Hence, the sense they make of the situation must include their own contributions to it. Yet they recognise that the situation, contrary to the intentions, may foil their projects and reveal new meanings. They do not act as though they had no relevant prior experiences. On the contrary, they attend to the peculiarities of the situation at hand, seek to discover the particular features of this problematic situation, and from this gradual discovery, frame the situation and design an intervention. The situation is uncertain, and there is a problem in finding and defining the problem.

[1] By "experient", Schön appears to mean an experimentor who is at the same time also a target or part of this experiment.

As the practitioners frame and reframe the problem by finding out the parts of the situation and their relationships, they suggest a direction for shaping the situation and seeing what belongs to the situation and what is outside. The practitioners then take the framed problem and conduct an experiment in the form of a planned or real act to discover what consequences and implications can be made to follow from the act. In order to see what can be made to follow from this framing and acting the practitioners try to adapt the situation once again to the frame. But the practitioners' acts also produce unintended changes that give the situation new meanings. The situation in a way talks back, and the practitioners must reframe the situation once again. The process spirals through stages of appreciation, action, and re-appreciation. The situation comes to be understood through the attempt to change it, and changed through the attempt to understand it.

Schön criticised "technical rationality" (1983, Chapter 3) according to which professional practice consists of the application of standardised scientific knowledge that is instrumental for problem solving. In contrast, Schön saw practitioners as involved in personal relationships, situations demanding action, and value conflicts. In this practice the practitioner develops knowledge that is often tacit and spontaneous, and some of this becomes routine. In order to break these routines, the practitioner needs the Reflection-in-Action. He criticises his means and ends, and becomes a researcher in the practice context. He will develop a practice oriented theory or rationale according to which he explains the situation and chooses his acts.

The word "theory" has here a special meaning, because according to Schön (1983, pp. 273-274), practitioners do not consider that they have formed a satisfactory account of phenomena in any practice situation until they have framed it in terms of their overarching theory, in terms of a rationale according to which they explain the situation and choose their acts. So theory has two intertwined meanings, first in action design and then in retrospective explanations and interpretations.

Reflective practice requires special skills in addition to the basic skills in communication. Raelin (2001) defines these skills as being, speaking, disclosing, testing, and probing. Being is central and pervasive, because it represents one's presence and vulnerability in creating a reflective climate among the participants. Vulnerability here means that one cannot rely on defending oneself against experience, for example in the case of a failure. In addition to these practical skills, the practitioners reporting to academic audience should master the basic research methods and be fluent in scholarly writing (Heiskanen 1995).

3. Our Experiences over the Years

In this section we present an overview of the history of the most important information system development projects of the University administration from the 1980's up to the present. It is out of the scope of this article to give a detailed account of all these projects and systems, but we have summarised them in Table 1. Some of the histories have been published earlier, and some of them are discussed here in order to give a vivid picture for the reader and background to our argumentation. The general level of success seems to be at least moderate, because the users of the major systems show rather high scores of satisfaction according to several surveys done with standard instruments of user information satisfaction measurement.

Our aim is to identify and discuss a learning "route" that is related to the development of a set reporting systems for the University community, beginning in 1990 and reaching to the present. This will be the major topic of this paper. The other systems make up the surroundings of this focal

process. The development history of the systems before 1990 forms the antecedent conditions (Newman and Robey 1992) of this reporting process.

We can divide the outcome of the projects in Table 1 into three classes: (1) successful, (2) problematic courses of action that eventually lead to success, and (3) outright failures. Success and failure of an information system can be defined in many ways (Lyytinen and Hirschheim 1987). We use the broad definition of Sauer (1993) who says that an IS development project is a failure when the management terminates it and stops funding. This definition is simple to use, but it admittedly does not take into account that the project may exceed its budget or schedule, or both. Budget and schedule overruns cannot be, however, considered always as failures, because IS development is also a learning process for the organisation. The case may be that during the process the developers find new ways to employ information technology, which may require more resources (budget overrun) or more time (schedule overrun). The extra resources spent may lead to a better solution and thus these overruns eventually lead to success. Below we briefly characterise some of the system according to the classification above.

Successful course of action seems to be related to two different types of projects. First group is made up of the adoptions of well-tried, standard application package (ACCOUNTING, Cash register management, ACM Fund transfer, Cost accounting). The second group consists of small-scale system development; 'small-scale' means here either limited functionality or low number of users, or both (Balance, Correspondence management, Capability databases). Problematic course of action seems to be related to either poor performance, i.e. slow response times, or poor functionality. This leads to the slowness of the work done with the help of these systems. PAYROLL represents the first subgroup. PERSONNEL, Budgeting (BU), OODI, and OODI Admissions suffered both performance and functionality problems.

Four projects were total failures. The oldest of them (ORDBAL) was an ambitious project that planned to implement a workflow of purchases, accounts payable, and property management into a single, streamlined system. This system was envisioned by a purchase clerk, but apparently she did not have enough support from her superiors. The visible reason to cancel this project was that the software had technical problems in network operations. The real reason was, however, that the first author considered it organisationally infeasible to continue the work, especially when the clerk had changed to work in another department of the University Administration. At hindsight, the termination of this project should have been done differently, not letting it fade away. This process, however, is out of the scope of this paper. A similar case of unprofessional project termination will be discussed in terms of UHMIS development below.

Two of the failures (HYTIX, MultiDoc) were small-scale systems that were planned to be used by the EDP office. The idea of HYTIX appeared to be too expensive to be fully developed. MultiDoc was a documentation and presentation system that contained descriptions of major administrative systems of the University. Its technical platform (ToolBook) became obsolete, because the Internet techniques (Mosaic, Netscape) were rapidly developing. The project was consequently terminated, because the benefits of this system were considered too low.

The most interesting failing case is UHMIS (University of Helsinki Management Information System) development. UHMIS was the first project in a series that eventually led to the successful data warehouse development project. All these projects were struggling how to deliver report data to the University community. The reporting systems development began in early 1990 with the software house CCC Software Professionals (Heiskanen and Similä 1992). The third party of this project was the University of Oulu because the staff of the project were four students; this type of

arrangement between Oulu University and a local software house (CCC was not the only firm of this type of cooperation) was a normal way of teaching IS development. Later a student continued the work as an employee of CCC. The role of the first author in this process was a member of the project board, responsible of how the IS development services are purchased.

At first there were difficulties to find out what the client (the planning office of the University) really wanted. Originally it was thought that a reporting prototype would have been the desired outcome from the first phase of the UHMIS project (spring 1990). However, suddenly during a project board meeting in March 1990 a flashing idea appeared that the outcome should be a system that would calculate performance indicators out of student records, personnel, and financial accounts. So the project was directed accordingly. The students produced a requirements specification report and made experiments with a brand new version of SAS software by the summer 1990. After that a contract was signed between the Helsinki University and CCC of further development.

The work continued seemingly well from autumn 1990 to summer 1992. CCC delivered pieces of software as agreed and the user representative signed documents that indicated that the delivery was as required. This, however, was only the surface. The resources of the University were very strained in both the user side as well as in the EDP side. So the management of the project was given to CCC and the involvement of University's own EDP personnel was negligible. Apparently the client did not test the deliveries enough. This came evident in autumn 1992. UHMIS produced erroneous statistics and its user interface was criticised to be clumsy when a broader audience looked at it. The planning office had engaged a new analyst to UHMIS work and she was very worried about the state of the software. She wrote the following note to the first author:

> "… I'm sending my first observations of UHMIS, preliminary feelings. Discussions will be a lot, but I thought to report which kind of problems we at least have to solve. I'm really afraid of the whole thing."

The client's EDP personnel now inspected the software, but no feasible remedy appeared. Several meetings were held, but the project was not officially terminated. In a way the work slipped to other areas that were related to management reporting and the development of UHMIS software was stopped. A reason to stop the development was that in 1993 the first OLAP-tools[2] appeared. With these tools the functions of UHMIS could be developed with a fraction of the resourced that would have been required with the earlier tools.

A typical reaction after a failure is to find who is guilty and who should compensate the damages. A possible target could be the software vendor. However, the first author in his practical role as the chief information systems officer of the University considered that recovering the losses (totalling to 100.000 Euros of software delivery payments) from the vendor would have been infeasible because of the two reasons. First, the client representative had accepted the deliveries and in the negotiations CCC representatives expressed their view that they have delivered just what the client had ordered. Second, CCC was performing well in other IS areas. Filing a court case would have impaired these relationships, and that would have been the only way to try to get the money back. The amount of losses was not so big that an unsure court case would have been the action to choose. Moreover, it seemed that the interest of the client unit towards this kind of system vanished little by little.

One of the areas related to the failed UHMIS was the development process for HURBS (Helsinki

[2] OLAP (On Line Analytical Processing) tools are easy-to-use programs for multidimensional data analysis.

University Reporting and Budgeting System) that began in 1991 following the Finnish State decision that a new management procedure should be installed in all state bureaux. The requirement was that the bureaux should move to a more objective-oriented management style, emphasising responsiveness to their clients and allowing more flexibility in the use of allowances.

The University made requirements analysis with the help of CCC during 1992 about the information systems that would be needed. In early 1993, after a bidding competition, the total HURBS project was divided into four subsystems: departmental account reporting (DEAC), personnel cost reporting (PERCOST), payroll prognoses (PAYPROG), and budgeting (BU). DEAC software was a slightly modified replica of the "official" accounting system. PERCOST development was started with CCC, but the system was finalised by University's own personnel. DEAC and PERCOST were put in use in 1993. BU's development was more time consuming and problematic, but that story (Heiskanen and Newman 1998) is out of the scope of this paper.

The HURBS specification project indicated that a data warehouse would be helpful in reporting. This was the fifth system that followed the HURBS specification. The data warehouse was planned to be an easy to use information repository that would get data from the transaction processing systems of University administration. Towards this end a database was designed, now using Oracle data base management software instead of SAS that was the tool for UHMIS. The user interface was developed using an OLAP-tool, and the system was named as WinUhmis. The development approach was also changed towards prototyping, i.e. delivering the software in small incremental pieces.

The reporting work continued in small steps. A more comprehensive approach was suggested by the internal auditor of the University in summer 1996. She wrote a memorandum that described an idea to develop an integrated reporting system. This memorandum was an indication of growing awareness of the lack of information that would be used for University management. However, the real needs of the users still seemed unclear and the possibilities of action in reporting development was impaired because EDP personnel were struggling with personnel and budgeting IS projects in 1996 and 1997.

In 1998 it seemed possible to establish a proper project for data warehouse development. It began as cooperation with two other universities, but quite soon the University of Helsinki continued its work irrespective of the two others. In this project, a very cautious way to proceed was chosen. The scope of work was decided to begin of accounting data that was familiar for the project leader, because his background was in economy administration. User participation was sought, but it appeared that no real input was coming from that side. The user management in the central administration even expressed in some occasions that this project would produce nothing useful. However, persistent work produced visible results when the user focus was changed from the university central administration level to department and faculty level. The first part (accounting data) of the data warehouse was operational in spring 2000. The result was good although project exceeded its schedule due to delays in software architecture decisions and difficulties with data transfer.

Second part of the data warehouse project, personnel and payroll, started in spring 2000. The project leader did not have the same experience in personnel sector as in accounting. Therefore deeper user participation was required. To EDP personnel it seemed that the Personnel department had some doubts of the usefulness of the data warehouse. At the same time there were severe performance problems in Personnel and payroll systems that affected also reporting. Even without performance problems there were shortcomings in reporting features of these systems. Therefore

users were ready to participate in data warehouse project that would result into a system that would remove load from the production systems and improve performance that way. Thus these users were more involved than their counterparts in accounting. Another difference to accounting was that a service provider runs the Payroll services. Also their participation was needed in the project.

Because dealing with personnel and payroll data was more complicated than was anticipated in the beginning of the project, project decided to handle first the payroll data. Project exceeded its time schedule. Main reasons were lack of service provider resources in some phases of the project, and use of the project leader and data warehouse specialist in other projects. Also the making of the conceptual schema took more time than was scheduled. Payroll information was included in the data warehouse in autumn 2001. Next step was to relate accounting and payroll information. This was done faster than was expected. Student information was also included in late 2001 and work continues to combine student data with accounting and payroll information.

So, the winding route of reporting development from 1990 up to now has led to a situation that can be considered as a success for the University. An additional indication of the success is that the Helsinki School of Economics and Business Administration, the leading Finnish university in this area, has adopted this system.

Project or system Time frame	**Description**	**Outcome and evaluation**
CERS 1985-1999	Centralised Student Records (VAX/VMS)	Successful development and implementation
DERS 1986-1999	Decentralised Student Records (PC/DOS)	Successful development and implementation
PERJOB 1986-1998	Personnel (VAX/VMS)	Successful development and implementation
BALANCE 1986-1995	Accounts payable, reporting, stand-alone system for departments (PC/DOS)	Successful development and implementation
ORDBAL 1988-1991	Purchase management and reporting (PC-network)	Failure because of organisational reasons
ACCOUNTING 1990 - >	On-line accounting package, own server	Successful adoptions of two generations of software packages
UHMIS 1990-1992	Performance indicator reporting of student records, economy, personnel, rooms (SAS)	Failure because of organisational and technical reasons; immature technology, underestimated development resources
HURBS specifications 1992-1993	Reporting and budgeting complex	Successful system architecture was developed
DEAC 1993 – 1998	A reporting replica of the accounting package for the departments	Successful adoption of the software package
PAYPROG 1993 ->	A software package to calculate payroll prognoses to be used in negotiations with labour unions	Successful adoption of the software package
PERCOST 1993 – 1998	An ORACLE database for reporting paid salaries to the departments	Successful development and implementation
Data warehouse prototyping 1993 – 1997	Constructing an Oracle database to support on line analytic processing (OLAP)	Positive small-scale experiences of the tools used
Budgeting	Budgeting and reporting systems	Eventually successful development and

1993 ->		implementation
HYTIX 1993-1994	Information systems documentation management (Windows)	Failure because of wrongly perceived user needs
MULTIDOC 1994-1996	Information systems description (ToolBook)	Failure because of wrongly perceived user needs; outdated technology
PAYROLL, from the 1960's onwards	Payroll services run by a service provider	Eventually successful adoption of a software package, performance problems
PERSONNEL 1994 ->	Personnel system integrated to the payroll, run by a service provider	Eventually successful development and implementation, severe performance problems
BU 1992 ->	Budgeting and reporting systems	Eventually successful development and implementation
Auditorium reservations 1990-1995	A system to manage centrally controlled auditoriums	Eventually successful development and implementation, but replaced just after implementation with another system because of organisational reasons
OODI Student records 1996 ->	Student Records, Uniface and Oracle	Eventually successful development and implementation, functionality and performance problems
OODI Student Admissions 1998 ->	Uniface and Oracle	Eventually successful development and implementation, functionality and performance problems
Capability databases 1994 ->	Lotus Notes based applications to publish data in WWW-pages about research and expertise	Successful development and implementation
Correspondence management 1997 ->	A Lotus Notes based register to master the official correspondence of the University	Successful modification of a software package and implementation
Cash register management 2000->	A software package to manage cash registers in various parts of the University; data transfer to the general ledger	Successful adoption of a software package
ACM Fund transfer 1997 ->	A software package to transmit account payable data to banks	Successful adoption of a software package
Cost accounting 1997 ->	A software package for cost accounting	Successful adoption of a software package
Data warehouse 1997 ->	Oracle database, accounting, payroll, personnel and student records data	Eventually successful development and implementation
Document base for objective negotia-tions (Lotus Notes) 1996 ->	A system for storing and transmitting the documents necessary in the annual objective negotiations	Successful deployment of a Lotus Notes database

Table 1. Overview of the projects and systems.

4. Organisational Learning out of the Reporting Development Process

In this section we analyse our experience from the learning point of view. The topic of organisational learning is in our case instrumental: how to successfully develop information systems for the University community. The main learners are the managers, project leaders, and systems analysts in the Information Systems Services of the University.

Our view of the definition of organisational learning is that of Robey, Boudreau and Rose (2000). Organisational learning means a process that enables the acquisition of, access to, and revision of organisational memory, thereby providing direction to organisational action. Organisational learning may be intentional or unintentional. We also use notions developed by Argyris and his colleagues when analysing our learning over the years (Argyris, Putman and Smith, 1987; Argyris and Schön 1978). In our case, the learning presented here is also an 'after-the-fact' interpretation of the history.

We frame the learning to consist of consecutive cycles. Each cycle begins with a reflective comprehension of the situation that demands action of the practitioner. Actions taken produce results that we call in the Schönian (Schön 1983) style as organisational back-talk, indicating that the results of the action may be different from the planned ones. Back-talk leads to reflection, which, in turn, is a predecessor of new actions. We illustrate our framing by presenting the history of the reporting systems development in a graphical format. Our interpretation of the history is in Figure 1 where the interplay between issues and events, problems, and action strategies is schematically presented. An issue or an event describes an occurrence that needs a reaction. Problem defines our comprehension of the situation. Strategy defines the way of solving the problematic situation.

Many large information systems evolve through generations. The time taken may be several decades, like in the classical case of Baxter and American Hospital Supply (Short and Venkatraman 1992), or in the cases studied within the Harward IS history project (Mason, McKenney and Copeland 1997). Thus the time taken to develop the Helsinki University data warehouse is not exceptionally long. It seems that in these long processes the learning cycles are also long. Our interpretation is that this history has contained three learning cycles. This is comparable to the learning cycles the first author has experienced when developing student information systems (CERS, DERS): four learning cycles during the years 1981-1993 (Heiskanen 1995).

UHMIS failure has understandable reasons that can be seen in the light of the current knowledge of the difficulties of data warehouse building. These projects require a great amount of work, especially data cleaning and transfer from the transaction processing systems to the data warehouse is extremely time consuming. This was unknown to the first author during the UHMIS project, and the IS community has learned it only recently. The second obvious reason was the lack of tools for OLAP.

Both the planning office (the client) and EDP Office lacked resources to tackle the UHMIS problems during the critical period from late autumn 1992 to summer 1993 in order to terminate the project in the right way. Several systems related to the new budgeting procedures were considered more urgent than the recovering of UHMIS. Moreover, the first author was engaged in his doctoral dissertation in 1992 and 1993. It was an easy way out to let the UHMIS project die slowly.

We could identify three learning cycles related to the development of reporting systems. The first one entails the failed UHMIS project. The second learning cycle begins with the HURBS specification project, proceeds developing provisional reporting systems with poor service level, and ends up to the beginning of the data warehouse development project. The third learning cycle consists of the data warehouse project.

The first learning cycle contains no direct learning, but the dilemmas of learning described by Argyris were clearly visible (Argyris, Putnam and Smith 1987, pp. 280-281). It has been possible only after several years to begin the detailed reflection over the UHMIS failure. Naming the Oracle

database and OLAP tools as WinUhmis can be seen as protective act in order to cope with frustration, anxiety, embarrassment and shame, the feelings Argyris relates to failure. When writing this paper, it has been very instructive to the first author to return to the data of this major failure. However, meanwhile he has been successful in other projects and in his way to analyse the experienced past. Without this success, the public analysis of a failure that he felt was at least partially his fault would have been much more difficult. At hindsight, it is hard to tell how much protection and how little reflection was present during and soon after the UHMIS failure, but anyway the project was silently buried and no immediate learning resulted.

The second learning cycle consists of the development process of the provisional reporting systems DEAC, PERCOST, PAYPROG as well as the data warehouse prototyping. This cycle means a very cautious approach and a modest level of ambition. The organisational back-talk seemed positive towards these actions. The poor service level of these systems was notified e.g. in the memorandum of the internal auditor. This led to the data warehouse project. The essence of the second learning cycle was some kind of 'wait-and-see' attitude.

The third learning cycle consists of the data warehouse development project. The main learning was that in this case the substance area expertise could be sought from various sources. First, as the project leader himself mastered accounting, he could represent the user side. So first the lack of support and input of the users could be balanced with a capable project leader and it was possible to proceed without a strong support and input of the user community. Later the faculty and department level users that were more interested than the central administration level users could replace the latter ones as the real user representatives. Deep knowledge of substance is vital to data warehouse and reporting development, and for accounting it could be secured from several sources. For the personnel and payroll data, the source of expertise was opened because the payroll and personnel experts realised that they had an indirect motivation to promote data warehousing: moving reporting from the production database to the data warehouse eliminated some performance problems of the production database.

It is possible to try to infer the theory-in-use of the history of our case using the principles suggested by Argyris (Argyris, Putnam and Smith 1987). In spite of the disappointing consequences over the years, it seems that the action strategies have aimed at delivering reporting systems to the University. The governing variables (ibid. p. 84) seem to include that an EDP professional should persistently act towards the evident goal of report development. The value of this variable, i.e. the amount of energy used to pursue this end, seems to vary over the years. During UHMIS, this variable got the form of sufficient funding but did not entail a strong hands-on involvement of University EDP personnel. After HURBS specification project the energy level can be deemed to be high both in monetary terms as well as personal involvement. In spite of high involvement, the action strategies seem to be cautious.

Another important governing variable seems to be the degree of user involvement. The normal textbook advice is to engage the users in the IS development work. In our process the degree of involvement cannot be deemed to be high. During UHMIS project the low input of the user community was one of the reasons of failure, but the project would have failed even with strong user support. Later the user involvement was higher. The HURBS specification project contained large amount of user interviews. The reporting systems that followed the specification were voluntary to use. It was possible to evaluate user reactions towards them and see whether they were used or not, and how satisfied the users were. The memorandum by the internal auditor was meant for public discussion within the user community. During the early data warehouse project (accounting data) the user involvement was lower again. During that time the EDP personnel were

confident that they could proceed without deep user involvement. When the first parts of the data warehouse would be operational, then it would be possible to gauge the user reactions and see how successful the project has been and plan the future according to the experiences.

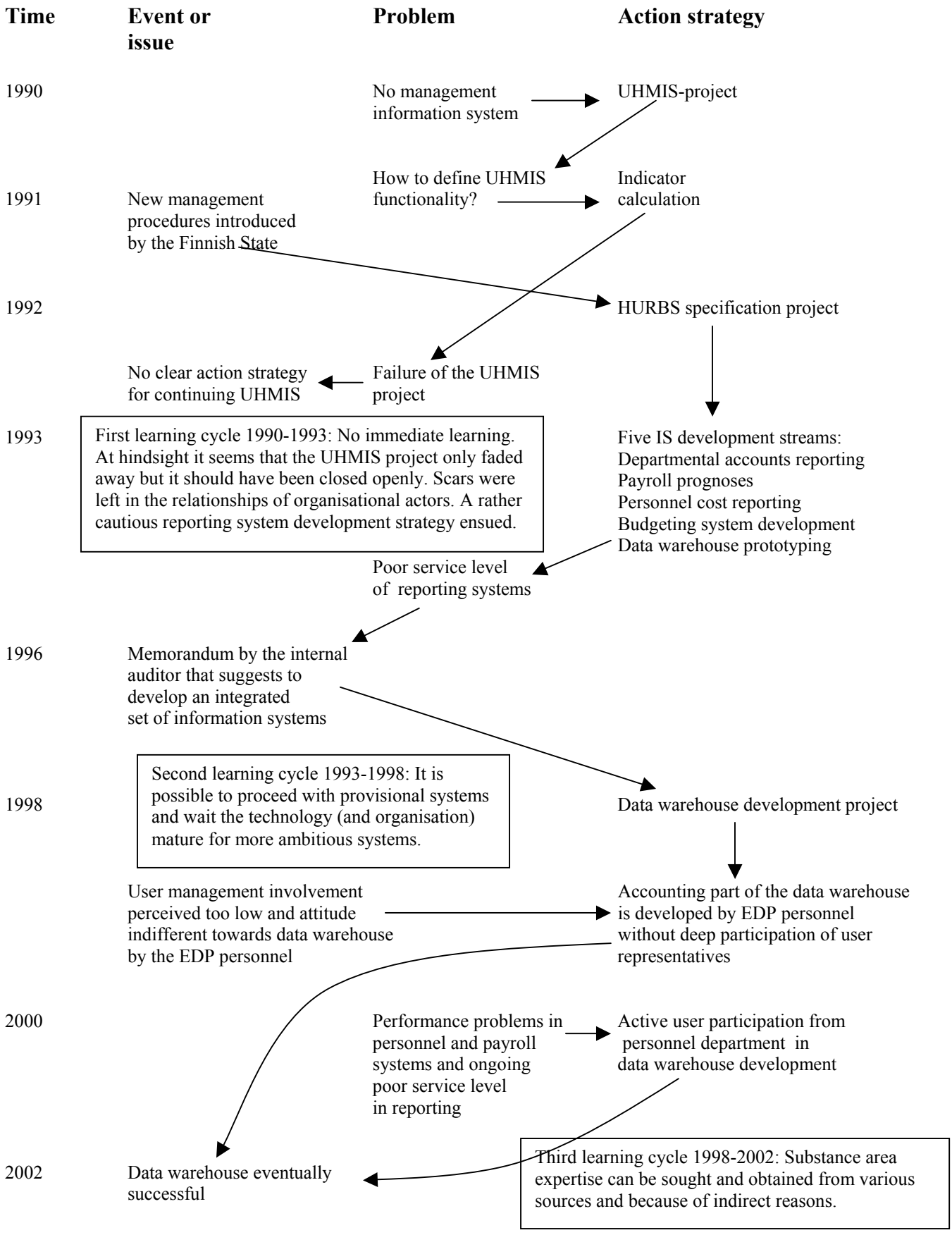

Table 1. The Learning cycles in reporting systems development.

It seems that the early theory-in-use held by the first author conforms the Model I type (Argyris, Putnam and Smith 1987), because the process contains features that seem to indicate unilateral control of the process by the EDP people and valid information was withheld when the UHMIS project was silently buried. The authors would like to think that over the years the theory-in-use would have come closer to Model II, which would mean that some learning has happened. Is it possible to find symptoms of this movement from Model I to Model II?

Model II theories-in-use (Argyris, Putnam and Smith 1987, p. 99) entail as governing variables valid information, free and informed choice, and internal commitment to the choice, and constant monitoring of its implementation. Action strategies are designed through the following principles: (1) design situations or environments where participants can be origins and can experience high personal causation, (2) tasks are controlled jointly, (3) protection of self is a joint enterprise and oriented toward growth, and (4) bilateral protection of others. Argyris et al. argue that in the long run the Model II theories-in-use lead to increased organisational effectiveness, but they also admit that the world more often than not continues to operate according Model I (ibid. p. 102).

We are apt to think that our history contains a movement toward Model II. During the HURBS specification project the users could freely express their views and guide the process. High personal causation (principle 1) and the joint control (principle 2) were arranged by the voluntary use of the reporting systems: the users could show their control through their amount of use. High use rate would indicate that the systems served user needs while low rate would indicate that improvements were needed. The amount of obedience to the principles of protection (3 and 4) is difficult to decide based on the data we have; that would entail a detailed observation of the development process meetings. Our data reveals only the great lines of development over the years.
It is a general feature of IS development that the EDP personnel cannot normally unilaterally control the projects, because the user community finally decides whether the IS products and services are of sufficient quality or not. These kinds of situations are in principle conducive for Model II behaviour between the users and the EDP professionals, because neither side is in unilateral control.

As a conclusion we can say that the reporting development process eventually found its way to success. Thus the organisation learned how to develop the reporting systems. Using the vocabulary of Robey, Boudreau and Rose (2000), the organisation could learn how to provide a direction of action that eventually was successful. At hindsight, it is possible to speculate about the possible explanations of how the three different development strategies of the respective learning cycles were devised. This relates the analysis of our history to the work of Nonaka and his colleagues (Nonaka and Takeuchi 1995; Nonaka, Toyama, and Konno 2000) about knowledge creation. According to them, tacit knowledge is conversed to explicit through abductive reasoning, i.e. through the use of figurative speech, metaphors and analogies.

Our attempt is to illustrate the birth of the three strategies with the help of three metaphors: linear strategy, logical incrementalism, and skunk works. These notions are from literature, but they have metaphorical power as such, because it is possible to squeeze the essence of the long process into the changes in the development strategy. Of course, it is a matter of taste to evaluate how much tacit knowledge and how much explicit knowledge were present when the respective acts were really designed. When looking back from our current point of view, the following interpretation seems to make sense.

During the UHMIS phase, the development strategy can be characterised to be linear (cf. Chaffee 1985) after the flashing idea of performance indicator calculation. Linear strategy means that strategy consists of integrated decisions, actions, and plans that are oriented towards setting and achieving viable organisational goals. A major assumption supporting this linear strategy model is that the environment is predictable. In our case, we could not predict the amount of work that would have needed. So the linear strategy failed. Evidently the previous experiences of the first author were affecting the decisions. Through a complicated process (Heiskanen, Newman and Simila 2000) a set of problematic projects had produced successful outcomes by 1990. That evidently gave faith that further projects would also succeed. At hindsight the UHMIS project seems very risky, but no real risk analysis was done.

The phase after HURBS specification can characterised as logical incrementalism (Quinn 1980). In logical incrementalism, the management gives a broad direction of action, but they do not set specific goals, on the contrary (Quinn 1980, p. 91-92):

> "[E]ffective top executives in major enterprises typically announce only a few broad goals from the top; they encourage their organizations to propose some; and they allow others to emerge from informal processes. They eschew the gimmickry of simplistic formal planning or MBO approaches for setting their major goals. Instead they tend to develop strategic goals through very complicated, largely political, consensus-building processes that are outside the structure of most formal management systems and frequently have no precise beginning or end."

The cautious incremental strategy produced symptoms of success, and it was possible to proceed to the next phase, the data warehouse project during the years 1998-2002. For this phase, we can coin the metaphor skunk works, originally introduced by Peters and Waterman (1982). Skunk works meant an operation that is performed outside the normal organisation by a competent but small group that contains persons with extra capability. This is slightly different in our case, because the work is performed within the normal organisation. The metaphor is valid in the sense that the development team was not tightly connected to the users but had freedom to proceed in their own way.

5. Discussion

In this section we discuss our reflective practice as a learning approach by comparing it with action learning and action science (as characterised by Raelin 1997a). Action learning (Raelin 1997a, p. 22, referring to the works of Revans (1980, 1982)) is a development approach, used in a group setting, that seeks to apply and generate theory from real organisational work situations. With the help of a facilitator, a series of presentations might be given on a designated theory or topic. During these presentations students might be asked to apply their prior and new knowledge to real projects. Not all organisational problems are solved or even are meant to be solved in action learning. Rather, the experience is designed to confront the learners with the constraints of organisational realities.

Action science (Raelin 1997, p. 23, referring to the works of Argyris and Schön) is an intervention approach to help learners increase their effectiveness in social situations through heightened awareness of the assumptions behind their actions and interactions. Key concepts are Model I and Model II action programs that are automatically activated in many interpersonal interaction situations. Model I aims at face saving, upset avoiding, and maintaining unilateral control. These kinds of reactions often produce self-reinforcing patterns that seal off self-discovery. Therefore the

action science facilitators try to engage the participants in Model II responses that allow for the exploration of interpersonal differences and mutual responsibility. The aim is to narrow inconsistencies between one's espoused theories (those characterising what we say that we do) and theories-in-use (those describe what we actually do). The goal of action science is to uncover our theories-in-use and distinguish between those that inhibit and those that promote learning. Schön (1983) prefers the term "reflection-in action". However, his approach (cf. Section 2) is close to action science.

We relate our reflective IS practice to action learning and action science in Table 2. The table is based on Raelin's (1997) conceptualisation on action technology criteria when comparing action learning and action science. We have added a column for Reflective IS Practice to the original table for this tripartite comparison. Action learning and action science are demanding points of comparison for our reflective IS practice, because they are established ways for organisational learning with a lot of experiences and large amount of publications. We have studied only a few IS histories in a single organisation and it is too premature to draw comprehensive conclusions. So our conjectures in Table 2 are very tentative. The general relationship between these three approaches can perhaps best be phrased that action learning and action science are proved approaches that can be included in IS projects, but they need competent facilitators. According to our point of view, it is out of question to try these methods only with the capabilities possessed by typical IS professionals.

In addition to the comparison in Table 2, we discuss three issues: definition of the learning situation, the length of the learning process, and the time direction of interpretations. It seems that there is a difference between, on the one hand, our reflective practice, and, on the other hand, action learning and action science in the definition of the learning situation. Action learning and action science see that the learning occasions are encounters of social life of the focal organisation in group settings. In our practice the learning is related to organisational back-talk, which in the development histories means events that are the organisational response towards the development acts. These responses may be related to the user reactions towards the proposed systems, or failures to deliver what is planned in the projects. By necessity, our histories contain situations that are suitable arenas for action learning or action science, but we have not taken advantage of those situations in the way action learners and action scientists would have done. The main substance of our learning is instrumental: how to act successfully in developing information systems to a specific user community.

The second issue is the length of the processes. Our processes have taken several years, while the cases typical of action learning seem to take months, action science interventions perhaps lasting somewhat longer. As already mentioned, typical IS development processes go through generations that may take a decade or two. The great length of IS histories is related to the economy of research and learning. A long process yields an enormous amount of data that may be cumbersome to analyse. There is, on the other hand, some positive with this, because we have had to gather only very little data for research purposes; normal project history documentation produces enough material for research.

Criteria	Action Learning	Action Science	Reflective IS Practice
Philosophical basis	Humanism and action research	Humanism and action research	Professional IS development work
Purpose	Behavioural change through reflection on real practices	Behavioural change through articulation of reasoning processes and improved public disclosure	Performance and behaviour improvement through reflection on real practices
Time frame of change	Short and mid-term	Long term	Very long term
Depth of change	Interpersonal and instrumental	Interpersonal and intrapersonal	Intrapersonal (professional), interpersonal
Epistemology	Placing theories into tacit experience	Making explicit tacit theories-in-use	Placing theories in interpretations of the past and formulation of the future
Nature of disclosure	Rational, making meaning from experience	Emancipatory, exploring the premises of beliefs	Rational, making inter-pretations of the experience
Ideology	Arising from intrinsic natural learning processes within the group	Subscribing to particularistic double-loop learning concerned with elicitation of mental models	Arising from the natural learning processes within the individual
Methodology	Processing there-and-then problems occurring within ones own work setting	Processing here-and-now reasoning, or on-line interactions	Processing there-and-then problems of own practice, designing acts based on past experience and current interpretations
Facilitator role	Passive, functioning as mirror to expedite group processing	Active, demonstrating and orchestrating on-line Model II learning skills	No external facilitator, the reflective practitioner is the facilitator
Level of inference	Low	High	May vary from low to high
Personal risk	Political, peer dissatisfaction or career derailment resulting from poor project performance	Psychological, exposure of personal defences and vulnerabilities	Political, from high to low, can be controlled by the practitioner
Organisational risk	Moderate, needs top management and supervisory management support	Heavy, requires all management levels to expose their assumptions	Low and can be controlled by the practitioner
Assessment	Project effectiveness, systemic change	Managerial effectiveness, systemic change	Professional effectiveness in series of projects
Learning level	Second order, challenging assumptions underlying practice interventions	Third-order, challenging premises underlying theories-in-use and underlying management's governing values	May vary from first order (methods of project management) to third order (challenging premises of past actions)

Table 2. Action technology criteria and distinctions between action learning, action science, and reflective IS practice (adapted from Raelin 1997a, p. 32)

The third issue seems to the time direction of theoretical interpretations. Action learning and action science seem to direct their vision to the future, while our approach is dominantly post hoc interpretation of systems histories. Originally (Heiskanen 1994, pp. 54-55), we thought that our approach would be according to the lines of action science (Argyris, Putnam and Smith 1987).

However, over the years it became apparent that the role of theory in action design was less than we expected. So theory testing which is a prominent feature in the work by Argyris has not been a visible feature in our research. Instead, more often we have used theorising only in retrospective interpretations. This is in line with the basic definition of reflective practice that means stepping periodically back to ponder self and others in the immediate organisational context.

6. Concluding remarks

In this working paper, we have described our conceptualisation of organisational learning that is related to reflective IS practice. Our main motive is to improve the capabilities of IS development through the analysis of the past. We gave a broad overview of nearly thirty projects and a more detailed account of how reporting systems were developed in three main phases. Each of these phases had its development strategy that was based on the experiences of the immediate past. We could relate the phases with respective learning cycles.

Our original plan was that we could have included issues of the creation of organisational memory already to this article. This appeared to be too ambitious a plan and the work will continue in further publications. Another topic for further work is to see how we can enlarge the amount of people engaged in this self-reflective learning process.

REFERENCES

Argyris, C., Putnam, R., and Smith, D. *Action Science*. San Francisco: Jossey Bass, 1987.

Argyris, C., and Schön, D. *Organizational Learning, A Theory of Action Perspective*. Reading: Addison-Wesley, 1978.

Ayas, K. and Zeniuk, N. "Project-based Learning: Building Communities of Reflective Practitioners", *Management Learning,* Volume 32, No.1, 2001 pp. 61-76.

Chaffee, E. E. "The Concept of Strategy: From Business to Higher Education." In J. C. Smart (Ed.), *Higher Education: Handbook of Theory and Research, I.* New York: Agathon, 1985.

Gill, T. G. "High-Tech Hidebound: Case Studies of Information Technologies That Inhibited Organizational Learning." *Accounting, Management and Information Technologies,* Volume 5, No 1, 1995, pp. 41-60.

Heiskanen, A. *Issues and Factors Affecting the Success and Failure of a Student Record System Development Process, a Longitudinal Investigation Base on Reflection-in-Action.* Doctoral Dissertation at the University of Tampere. Helsinki: the University of Helsinki, 1994.

Heiskanen, A. "Reflecting over a Practice, Framing Issues for Scholar Understanding." *Information Technology and People*, Volume 8, 1995, pp. 3-18.

Heiskanen, A. and Newman, M. "Bridging the Gap Between Information Systems Research and Practice: The Reflective Practitioner as a Researcher". *Proceedings of the Eighteenth International Conference on Information Systems,* Kumar, K. & DeGross, J.I. (Eds.), Atlanta, Georgia, December 15-17, 1997, pp. 121 - 131.

Heiskanen, A. and Newman M. "The Dynamics of IS Procurement, Case Study of a Budgeting and Financial Reporting System". *Proceedings of the 6th European Conference on Information Systems*, Baets, W. R. J. (Ed.), Aix-En-Provence, 3-6.6.1998, pp. 839-852.

Heiskanen, A., Newman, M. and Similä, J. "The Social Dynamics of Software Development". *Accounting, Management and Information Technologies*, Volume 10, 2000, pp. 1-32.

Heiskanen, A., and Similä, J. "Gatekeepers in the Action Structure of Software Contracting: A Case Study of the Evolution of User-Developer Relationships." *ACM Computer Personnel*, Volume 14, 1992, pp. 30-44.

Lyytinen, K. & Hirschheim, R. "Information Systems Failures - A Survey and Classification of the Empirical Literature", *Oxford Surveys in Information Technology*, Volume 4, 1987, pp. 257-309

Lyytinen, K. and Robey, D. "Learning Failure in Information Systems Development." *Information Systems Journal,* Volume 9, 1999, pp. 85-101.

Mason, R. O., McKenney, J. L., and Copeland, D. G. "Developing an Historical Tradition in MIS Research." *MIS Quarterly,* September 1997a, pp. 257-278.

Newman, M. and Robey, D. "A Social Process Model of User-Analyst Relationships", *MIS Quarterly*, June 1992, pp. 249-266.

Nonaka, I., and Takeuchi, H. *The Knowledge-Creating Company*. New York: Oxford University Press, 1995.

Nonaka, I., Toyama, R., and Konno, N., "SECI, Ba and Leadership: a Unified Model of Dynamic Knowledge Creation." *Long Range Planning*, Volume 33, 2000, no 1, pp. 5-34.

Peters, T. J. and Waterman, R. H. *In Search of Excellence, Lessons from America's Best-Run Companies.* London: Harper&Row, 1982.

Quinn, J. B. *Strategies for Change*, Logical Incrementalism, Irwin, USA, 1980

Raelin, J. A. "Action Learning and Action Science: Are They Different?" *Organizational Dynamics*, Volume 26, 1997a, No 1, pp. 21-34.

Raelin, J. A. "A Model of Work-Based Learning", *Organization Science,* Volume 8, 1997b, No. 6, pp. 563-578.

Raelin, J. A. "Public Reflection as the Basis of Learning", *Management Learning,* Volume 32, 2001, No. 1, pp. 11-30.

Revans, R. *Action Learning.* London: Blond & Briggs, 1980.

Revans, R. *The Origins and Growth of Action Learning.* Bromley: Chartwell Bratt, 1982.

Robey, D., and Newman, M. "Sequential Patterns in Information Systems Development: An Application of a Social Process Model." *ACM Transactions of Information Systems*, Volume 14, January 1996, pp. 30-63.

Robey, D., Boudreau, M-L., and Rose, G. M. "Information technology and organizational learning: a review and assessment of research." *Accounting, Management and Information Technologies*, Volume 10, 2000, pp. 125-155.

Sauer, C. *Why Information Systems Fail: A Case Study Approach*. Henley-On-Thames: Alfred Waller Limited, 1993.

Schön, D. *The Reflective Practitioner, How Professionals Think in Action.* New York: Basic Books, 1983.

Schön, D. *Educating the Reflective Practitioner, Toward a New Design for Teaching and Learning in the Professions*. San Francisco: Jossey Bass, 1987.

Seibert, K. W. "Reflection-in-Action: Tools for Cultivating On-the-Job learning Conditions." *Organizational Dynamics,* Winter 1995, pp. 54-65.

Short, J. E. and Venkatraman, N. "Beyond Business Process Redesign: Refining Baxter's Business Network." *Sloan Management Review*, Fall 1992, pp. 7-21.