**Learning cycles, organizational back talk, and the persistence of ...**
Heiskanen, Ari;Assinen, Pauli
*Knowledge and Process Management;* Jul/Sep 2003; 10, 3; ProQuest
pg. 183

Knowledge and Process Management Volume 10 Number 3 pp 183–193 (2003)
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/kpm.177

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

Ari Heiskanen[1]* and Pauli Assinen[2]

[1]*Department of Information Processing Sciences, University of Oulu, Finland*
[2]*Information Systems Services, IT Department, University of Helsinki*

Learning experiences from twenty-seven 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 was chosen for more detailed inspection. Organizational learning was conceptualized to consist of cycles during which the organizational reactions towards development actions reveal the successfulness of each project. In a retrospective analysis the authors found 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. Copyright © 2003 John Wiley & Sons, Ltd.

## 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 *et al.*, 2000): first, implementing an IS necessarily entails organizational learning, and, second, information technology can be designed to be part of the organizational memory that supports 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 our analysis of the histories of twenty-seven IS development projects in the administration of the University of Helsinki. The time frame covers the years 1985–2001. Our aim is to learn what has been successful in this context, and why, and why some projects failed. Our study is related to the previous work of the first author as the chief information systems officer of the University of Helsinki (Heiskanen, 1995; Heiskanen and Similä, 1992; Heiskanen and Newman, 1997, 1998; Heiskanen *et al.*, 2000). As the first author now has moved, at least temporarily, to an academic position in another university, this article can also be seen as an attempt to transfer the expertise of the first author to the second one who now continues in the capacity of chief information systems officer and the head of the information systems services unit of the university (the EDP Office).

We investigate the possibilities of putting our direct experience from practice into a form that makes sense to both academic and practical

*Correspondence to: Ari Heiskanen, Pähkinänsärkijä 3 C 16 FIN-01710 Vantaa, Finland. E-mail: Ari.Heiskanen@Oulu.fi

Copyright © 2003 John Wiley & Sons, Ltd.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission. www.manaraa.com

audiences (Heiskanen and Newman, 1997). Our task is what Nonaka and his colleagues (Nonaka and Takeuchi, 1995; Nonaka *et al*., 2000) call knowledge conversion from unarticulated practice to explicit knowledge. Our approach is not a rare one, because there exist other examples of how to reflect on 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 on the actions of oneself 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 questions the presuppositions attending the problem. The timing of reflection may be anticipatory, contemporaneous, or retrospective. Originally, Schön (1983, p. 163) characterized the work of design as a reflective conversation with the situation where the practitioner functions as an agent and experient that appears to mean an experimenter who is at the same time also a target or part of this experiment. He coined the term 'reflection-in-action' to describe this.

## 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 mid 1980s up to the present. It is beyond the scope of this article to give a detailed account of all these projects and systems, but we have summarized them in Table 1. Some of the histories have been published earlier, and some of

*Table 1 Overview of the projects and systems*

| Project or system time frame | Description | Outcome and evaluation |
|---|---|---|
| CERS 1985–1999 | Centralized student records (VAX/VMS) | Successful development and implementation |
| DERS 1986–1999 | Decentralized student records (PC/DOS) | Successful development and implementation |
| PERJOB 1986–1998 | Personnel (VAX/VMS) | Successful development and implementation |
| BALANCE 1986–1995 | Accounts payable, reporting, standalone system for departments (PC/DOS) | Successful development and implementation |
| ORDBAL 1988–1991 | Purchase management and reporting (PC-network) | Failure for organizational 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 organizational 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 salaries paid 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 (BU) 1993 → | Budgeting and reporting systems | Eventually successful development and 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 1960s 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 |

*Continues*



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission. www.manaraa.com

*Table 1 Continued*

| | | |
|---|---|---|
| Auditorium reservations 1990–1995 | A system to manage centrally controlled auditoriums | Eventually successful development and implementation, but replaced just after implementation with another system for organizational reasons |
| Oodi student records 1996 → | Student records system (Uniface and Oracle) | Eventually successful development and implementation, functionality and performance problems |
| Oodi student admissions 1998 → | Student admission system (Uniface and Oracle) | Eventually successful development and implementation, functionality and performance problems |
| Capability databases 1994 → | Applications to publish data in WWW-pages about research and expertise (Lotus Notes) | Successful development and implementation |
| Correspondence management 1997 → | A register to master the official correspondence of the university (Lotus Notes) | 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 → | Integrated database, accounting, payroll, personnel and student records data (Oracle) | Eventually successful development and implementation |
| Document base for objective negotiations 1996 → | A system for storing and transmitting the documents necessary in the annual objective negotiations (Lotus Notes) | Successful deployment of a Lotus Notes database |

them are discussed here in order to give the reader a vivid picture and provide background to our argumentation. The general level of success seems to be at least moderate, because only four systems were outright failures and the users of the major systems have given rather high scores for their satisfaction with them according to several surveys done with standard instruments of user information satisfaction measurement.

Our major task is to identify and discuss a learning 'route' that is related to the development of a set of reporting systems for the university community, beginning in 1990 and reaching the present. 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 its 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 organization. The developers may 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.

Four projects were total failures. The oldest of them (ORDBAL) was an ambitious project to implement the 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 for cancelling this project was that the software had technical problems in network operations. The real reason however, was, that the first author considered it organizationally unfeasible to continue the work, especially after the clerk had



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission. www.manaraa.com

moved to another department of the university administration. In hindsight, the termination of this project should have been done differently, not just letting it fade away. 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 Internet techniques (Mosaic, Netscape) were rapidly developing. The project was consequently terminated, because the benefits of this system were considered to be too low.

The most interesting failure case is UHMIS (University of Helsinki Management Information System) development. UHMIS was the first project in a series that eventually led to a successful data warehouse development project. All these projects were struggling with the problem of 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 associated with 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 as a member of the project board, responsible for how the IS development services were purchased.

At first there were difficulties in finding 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 out of the blue there was the proposal 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 experimented with a brand new version of the SAS software by summer 1990. After that a contract was signed between Helsinki University and CCC for 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 superficially true. The resources of the university were very strained in both the user and the EDP sides. So the management of the project was given to CCC and the involvement of the university's own EDP personnel was negligible. Apparently the client did not test the delivered programs well enough. This became evident in autumn 1992. UHMIS produced erroneous statistics and its user interface was criticized as being clumsy when a broader audience looked at it. The planning office had engaged a new analyst for the 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 what kinds of problems we at least have to solve. I'm really afraid about 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 the UHMIS software was stopped. The reason for stopping the development was that in 1993 the first OLAP-tools appeared. OLAP (On Line Analytical Processing) tools are easy-to-use programs for multidimensional data analysis. With these tools the functions of UHMIS could be developed at a fraction of the resources that would have been required with the earlier tools.

One of the areas related to the failed UHMIS was the development process for HURBS (Helsinki University Reporting and Budgeting Systems) 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, emphasizing 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). The DEAC software was a slightly modified replica of the 'official' accounting system. PERCOST development was started with CCC, but the system was finalized by the University's own



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission. www.manaraa.com

personnel. DEAC and PERCOST were taken into use in 1993. BUs development was more time consuming and problematic, but that story (Heiskanen and Newman, 1998) is beyond 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 the university administration. Towards this end a database was designed, now using Oracle database management software instead of SAS, which was the tool used for UHMIS. The user interface was developed using an OLAP-tool, and the system was named 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 her idea of how to develop an integrated reporting system. This memorandum was an indication of growing awareness of the lack of information that could 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 that it could be 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 with accounting data that was familiar to the project leader, because his background was in financial management. User participation was sought, but it appeared that no real input was coming from that side. The user management in the central administration even said that this project would produce nothing useful. Quite probably this opinion was because the progress of the development work had been so slow. However, persistent work produced visible results when the user focus was changed from the university central administration to department and faculty level. The first part (accounting data) of the data warehouse was operational in spring 2000. The result was good although the project exceeded its schedule due to delays in software architecture decisions and difficulties with data transfer.

The second part of the data warehouse project, personnel and payroll, started in spring 2000. The project leader did not have the same experience in the personnel sector as he had in accounting. Therefore deeper user participation was required. To EDP personnel it seemed that the Personnel Department had some doubts about the usefulness of the data warehouse. At the same time there were severe performance problems in personnel and payroll systems that also affected reporting. In addition to performance problems there were shortcomings in the reporting features of these systems. Therefore, users were ready to participate in the data warehouse project that would remove some of the load from the production systems and in this way improve performance. Thus, these users were more involved than their counterparts in accounting. Another difference between the personnel and accounting sectors was that a service provider runs the payroll services. Therefore, their participation was also needed in the project.

Because dealing with both personnel and payroll data was more complicated than was anticipated, the project decided to handle the personnel data first. The work took more time than had been anticipated because of the lack of service-provider resources, and because the project leader and the data warehouse specialist had other duties. Payroll information was included in the data warehouse in autumn 2001. The next step was to relate accounting and payroll information. Student information was also included in late 2001 and the work is now continuing to combine student data with accounting and payroll information.

## ORGANIZATIONAL LEARNING OUT OF THE REPORTING DEVELOPMENT PROCESS

In this section we analyse our experiences from the learning point of view. The aim of organizational learning in our case is instrumental: how to successfully develop information systems for the university community. The main audience for learning are the managers, project leaders, and systems analysts in the Information Systems Services Unit of the University. Organizational learning involves a process that enables the acquisition of, access to, and revision of organizational memory, thereby providing direction to organizational action (Robey *et al.*, 2000).

We framed the learning to consist of consecutive cycles. Each cycle began with a reflective comprehension of the situation that demanded the action of the practitioner. Actions taken produced results



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission. www.manaraa.com

that we called in the Schönian (Schön, 1983) style organizational 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. The problem defines our comprehension of the situation. The strategy defines the way the problematic situation is solved.

Many large information systems evolve through generations. The time taken may be several decades (e.g. Short and Venkatraman, 1992; Mason *et al*., 1997). Thus the time taken to develop the 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 experienced when developing student information systems (CERS, DERS): four learning cycles during the years 1981–1993 (Heiskanen, 1995).

The failure of UHMIS has understandable reasons that can be seen in the light of the current understanding of the difficulties of data warehouse building. These projects require a great amount of work, especially the data cleaning and transfer from the transaction processing systems to the data warehouse is extremely time consuming. This was not apparent to the first author during the UHMIS project, and the IS community has only recently learned it. The second obvious reason was the lack of tools for OLAP.

Both the planning office (the user unit) and the 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 finishing his doctoral dissertation in 1992 and 1993. Thus letting the UHMIS project die slowly was an easy way out.

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 with the developing of provisional reporting systems with poor service level, and ends up with 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 *et al*., 1987, pp. 280–281). It has been possible only after several years to begin detailed reflection over the UHMIS failure. When writing this paper, it has been very instructive for the first author to return to the data of this major failure. It seems now obvious that excessive protection, instead of open reflection, inhibited organizational learning out of two failing projects, UHMIS and ORDBAL.

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 involved a very cautious approach and a modest level of ambition. The organizational 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 lesson learnt here was that in this case the specialist expertise could be sought from various sources. First, as the project leader himself mastered accounting, he could represent the user side. So, at 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 the strong support and input of the user community. Later faculty and department level users that were more interested than the central administration level users could replace the latter as the real user representatives. A deep knowledge of substance is vital for data warehouse and reporting development. For accounting it could be secured from several sources. For the personnel and payroll data, the source of expertise was readily available because the payroll and personnel experts realized that they had an indirect motivation to promote data warehousing because it eliminated performance problems.

In conclusion we can say that the reporting development process eventually found its way to a successful end. Thus the organization learned how to develop the reporting systems. Using the vocabulary of Robey *et al*. (2000), the organization could learn how to provide directions of actions that eventually led to success. In hindsight, it is possible to speculate about 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 *et al*., 2000) about knowledge creation. According to



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission. www.manaraa.com

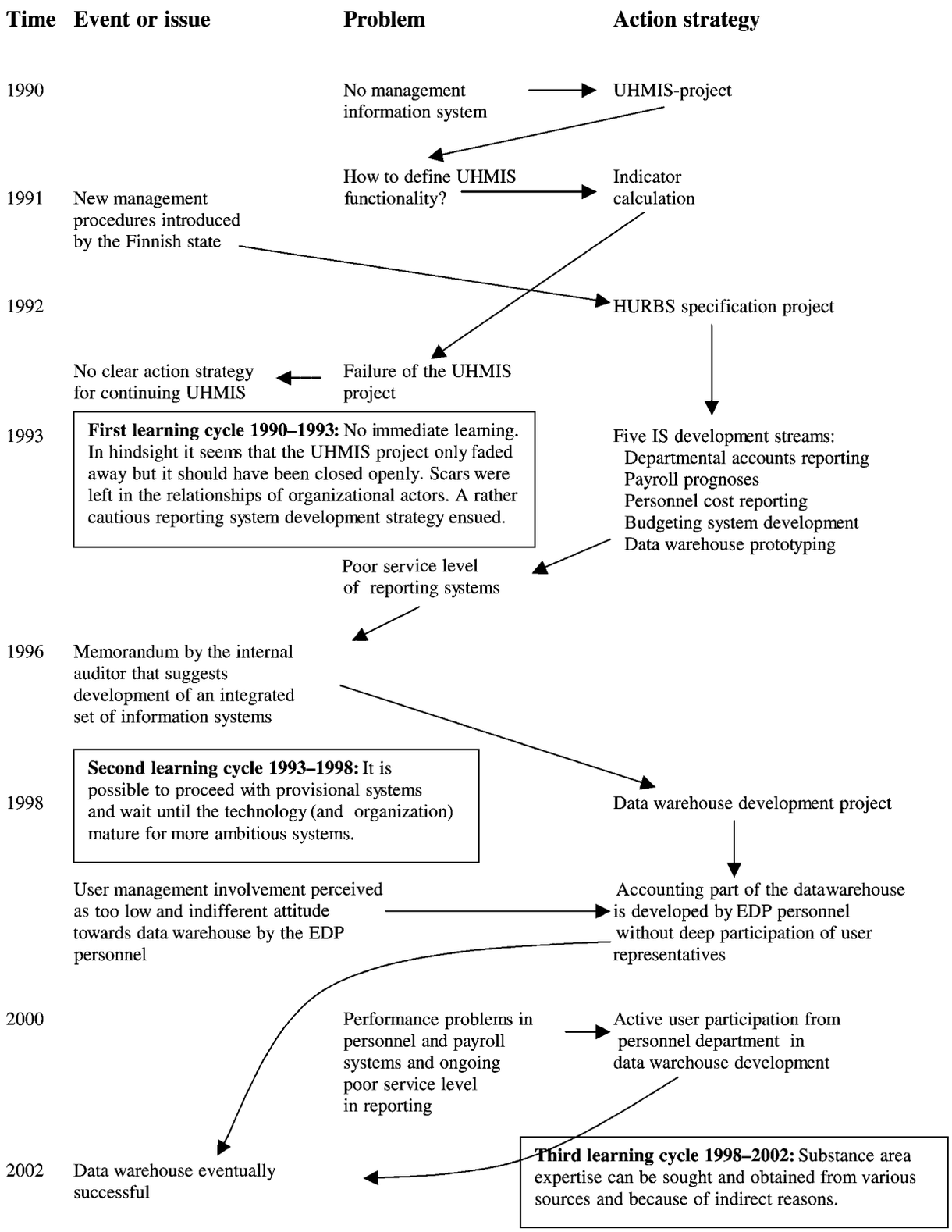


*Figure 1 The learning cycles in reporting systems development*

them, tacit knowledge is converted to explicit knowledge through abductive reasoning, i.e. through the use of figurative speech, metaphors and analogies.

We illustrate the birth of the three strategies with the help of three metaphors: linear strategy, logical incrementalism, and skunk works. These notions are from the literature, but they also have metaphorical



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission. www.manaraa.com

power as such. 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 characterized as being linear (cf. Chaffee, 1985) after the flashing idea of performance indicator calculation. A linear strategy means that strategy consists of integrated decisions, actions, and plans that are oriented towards setting and achieving viable organizational 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 been needed. So the linear strategy failed. Evidently the previous experiences of the first author were affecting the decisions, because through a complicated process (Heiskanen *et al.*, 2000) a set of problematic projects had produced successful outcomes by 1990. This evidently gave hope that further projects would also succeed. In hindsight the UHMIS project seems very risky, but no real risk analysis was done.

The phase after HURBS specification can be characterized as logical incrementalism (Quinn, 1980). In logical incrementalism, the management gives a broad direction of action, but, however, they do not set specific goals (Quinn, 1980, pp. 91–92):

> [E]ffective top executives in major enterprises typically announce only a few broad goals from the top; they encourage their organisations to propose some; and they allow others to emerge from informal processes. They eschew the gimmickry of simplistic formal planning or MBO [Management-by-objectives] 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. 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 organization 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 organization. However, the metaphor is valid in the sense that the development team was not tightly connected to the users but they were to proceed as they wished.

The learning cycles are here mainly seen from the point of view of the first author, the chief information systems officer of the university, and his unit. It is possible to infer that their theory-in-use (Argyris *et al.*, 1987) over the years has been persistent motivation to resolve the reporting system problems via different action strategies. They learnt that the amount of user involvement may be different in different phases of development. They also seemed to learn how to proceed when the users lacked time or motivation to participate.

Wider organizational learning also developed during this process. It took the shape of improving project management techniques. The role of these techniques was, however, non-existent during the first learning cycle and negligible during the third learning cycle. During the first learning cycle they had not been developed, and as the data warehouse development took the form of skunk works, the procedures were applied only minimally during the third learning cycle.

Formal project planning and control methods were introduced during the second learning cycle, in parallel with the development of the systems. This methodological work helped to guide the development work, but it also made it even more apparent that there was an urgent lack of experienced IS development work force, both user representatives and EDP personnel. On many occasions the user directors realized that they had to release their most competent personnel from other duties to IS development work. Sometimes they succeeded, but sometimes it seemed that they paid only lip service to project plans. Unfortunately space does not allow for the further discussion of organizational level learning here, so we will have to save it for later publications.

## DISCUSSION

In this section we discuss our reflective practice as a learning approach by comparing it with action learning and action science (as characterized by Raelin, 1997). Action learning (Raelin, 1997, 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 organizational 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 organizational 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 organizational realities.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission. www.manaraa.com

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 to save face, avoiding upsetting others, and maintain unilateral control. These kinds of reactions often produce self-reinforcing patterns that seal off self-discovery. Therefore 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 characterizing what we say that we do) and theories-in-use (those describing 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.

We relate our reflective IS practice to action learning and action science in Table 2. The table

*Table 2 Action technology criteria and distinctions between action learning, action science, and reflective IS practice*

| 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 for formulation of the future |
| Nature of disclosure | Rational, making meaning from experience | Emancipatory, exploring the premises of beliefs | Rational, making interpretations 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 one's 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 a 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 |
| Organizational 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) |

Adapted from Raelin 1997, p. 32.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission. www.manaraa.com

is based on Raelin's (1997) conceptualization of 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 organizational learning with a lot of experience gained and a large number of publications. We have studied only a few IS histories in a single organization and it is too premature to draw comprehensive conclusions. So our conjectures in Table 2 are only tentative. The general relationship between these three approaches can perhaps best be worded thus: action learning and action science are proven 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 with only 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 claim that the learning occasions are encounters in the social life of the focal organization in group settings. In our practice the learning is related to organizational back-talk, which in the development histories mean events that are the organizational response to the development acts. These responses may be related to the user reactions to 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 for 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 last somewhat longer. As already mentioned, typical IS development processes go through generations that may take a decade or two. These generations are related to the organizational role of the information systems. The systems support vital functions of the organization and they are periodically renewed in order to take the advantage of new information technology.

The third issue seems to be the time direction of theoretical interpretations. Action learning and action science seem to direct their vision to the future, while our approach in this process has been dominantly post hoc interpretation of system history. Originally (Heiskanen, 1994, pp. 54–55), we thought that our approach would be according to the lines of action science (Argyris *et al.*, 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 this case. Instead, we have used theorizing in retrospective interpretations. It is possible to speculate that through the retrospective analysis we could make the tacit theories of action more explicit.

## CONCLUDING REMARKS

We have described our conceptualization of organizational learning that is related to reflective IS practice. Our main motive has been 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 a development strategy that was based on experiences gained in the immediate past. We could relate the phases to respective learning cycles. Our plan for future work is that we would like to discuss the issues of the creation of organizational memory for the Information Systems Services Unit of the university. Another topic for further work is to see how we can enlarge the number of people engaged in this self-reflective learning process and how to enhance comprehensive organizational learning.

## REFERENCES

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

Ayas K, Zeniuk N. 2001. Project-based learning: building communities of reflective practitioners. *Management Learning* **32**(1): 61–76.

Chaffee EE. 1985. The concept of strategy: from business to higher education. In *Higher Education: Handbook of Theory and Research, I*, Smart JC (ed.). Agathon: New York.

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



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission. www.manaraa.com

Heiskanen A. 1995. Reflecting over a practice, framing issues for scholar understanding. *Information Technology and People* **8**(4): 3–18.
Heiskanen A, Newman M. 1997. 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 JI (eds). Atlanta, Georgia, December 15–17: 121–131.
Heiskanen A, Newman M. 1998. The dynamics of IS procurement, case study of a budgeting and financial reporting system. *Proceedings of the 6th European Conference on Information Systems*, Baets WRJ (ed.). Aix-En-Provence, 3–6 June 1998: 839–852.
Heiskanen A, Newman M, Similä J. 2000. The social dynamics of software development. *Accounting, Management and Information Technologies* **10**(1): 1–32.
Heiskanen A, Similä J. 1992. Gatekeepers in the action structure of software contracting: a case study of the evolution of user-developer relationships. *ACM Computer Personnel* **14**(1&2): 30–44.
Lyytinen K, Hirschheim R. 1987. Information systems failures—a survey and classification of the empirical literature. *Oxford Surveys in Information Technology* **4**: 257–309.
Lyytinen K, Robey D. 1999. Learning failure in information systems development. *Information Systems Journal* **9**(2): 85–101.
Mason RO, McKenney JL, Copeland DG. 1997. Developing an historical tradition in MIS research. *MIS Quarterly* **21**(3): 257–278.
Newman M, Robey D. 1992. A social process model of user– analyst Relationships. *MIS Quarterly* **16**(2): 249–266.
Nonaka I, Takeuchi H. 1995. *The Knowledge-Creating Company*. Oxford University Press: New York.
Nonaka I, Toyama R, Konno N. 2000. SECI, Ba and leadership: a unified model of dynamic knowledge creation. *Long Range Planning* **33**(1): 5–34.
Peters TJ, Waterman RH. 1982. In *Search of Excellence, Lessons from America's Best-Run Companies*. Harper & Row: London.
Quinn JB. 1980. *Strategies for Change, Logical Incrementalism*. Irwin: USA.
Raelin JA. 1997. Action learning and action science: are they different? *Organizational Dynamics* **26**(1): 21–34.
Raelin JA. 2001. Public reflection as the basis of learning. *Management Learning* **32**(1): 11–30.
Revans R. 1980. *Action Learning*. Blond & Briggs: London.
Revans R. 1982. *The Origins and Growth of Action Learning*. Chartwell Bratt: Bromley.
Robey D, Newman M. 1996. Sequential patterns in information systems development: an application of a social process model. *ACM Transactions of Information Systems* **14**(1): 30–63.
Robey D, Boudreau M-L, Rose GM. 2000. Information technology and organizational learning: a review and assessment of research. *Accounting, Management and Information Technologies* **10**(2): 125–155.
Sauer C. 1993. *Why Information Systems Fail: A Case Study Approach*. Alfred Waller Limited: Henley-On-Thames.
Schön D. 1983. *The Reflective Practitioner, How Professionals Think in Action*. Basic Books: New York.
Short JE, Venkatraman N. 1992. Beyond business process redesign: refining Baxter's business network. *Sloan Management Review* **34**(1): 7–21.



Reproduced with permission of the copyright owner. Further reproduction prohibited without permission. www.manaraa.com