The Adoption of new Application Development Tools by IT Professionals from the Viewpoint of Organisational Learning

Torsti Rantapuska
Faculty of Business Studies, Lahti Polytechnic, Finland
torsti.rantapuska@lamk.fi

Abstract: Productivity and innovativeness of information work is becoming an important issue among information workers. This paper explores the working and learning of IS professionals when adopting new application development tools. I study how the IS professionals work, communicate, think through problems, and learn by way of getting work done. I also analyse the changes that the adoption causes to the individual style of working. The research questions are formulated as follows: 1) what contributes to the effective use of IT tools? 2) How does the adoption of new tools affect the individual working methods? The research is based on interviews of fourteen young professionals who have recently started using a new application development tool. The interviews have been conducted in their working places. The focus is on learning at work. Special attention is paid to the initial motivation of the innovation, to knowledge acquisition, and to communication with their team members during the problem solving process. According to the findings, the IS professionals' working style is personal and context-oriented. As learners they do not interact with their peers and do not use systematic working methods too much. The Internet and help systems are used as the basis of group interaction and source of knowledge more likely than colleagues and textbooks. The systematic orientation of working practice is limited to the context at hand. At the end of the study, the results are discussed and recommendations are proposed to improve the software process.

Keywords: software process innovations, organisational learning, adoption, individual learning styles

1. Introduction

Much attention is paid to software process improvement (SPI). The ideas behind this research have mainly concentrated on the technical and methodological aspects (Fuggetta 2000). The SPI is usually regarded as a process of sequential practices of software engineering, implementation, and management (Wang & Bryant 2002). With respect to the importance of rigorous planning and management of a software project, it has been stated that there are many non-technical problems, which should be handled first (Humphrey et al 1989). The success of the software project depends very much on the social factors, such as group-interaction (Markulis P et al 1998), culture (Seng et al 2002), commitment (Abrahamsson 2002), and learning related activities (Gallivan 1995). It has been shown that more than 80% of learning is implicit and tacit in nature (Eraut 2000).

This paper analyses how learning takes place in the context of software tools adoption and how the adoption affects the working practices. The findings show a motivated adoption of new tools, but not so much of systematic working methods. The IS professionals rely on their individual skills and knowledge more likely than on organisational standards. They use electronic based sources, like the internet and ad hoc help systems. The use of peers and text sources remain minimal.

2. Research Approach

2.1 Research goals and questions

The aim of this study is to describe the adoption, learning and usage of new software tools among IT-professionals. In addition, I analyse the changes caused by the adoption to the developers' working methods and practices. The focus is on the individual and group levels. I am also interested in how working people communicate, acquire knowledge, think through problems, and learn by getting the work done.

The research questions are formulated as follows:

1. What contributes to effective use of IT tools?
   - What is the adoption process of new tools like?
   - How is the learning process of new tools?
   - How does group process affect the use of tools?
2. How does the adoption of new tools affect the individual working methods?

2.2 Theoretical assumptions

In recent studies working and learning are discussed from the standpoint of organisational learning (Nonaka 1994, Sunassee & Sewry 2003, Senge et al 2002, Senge 1990a, 1990b, Brown & Duguid P. 1991, 2001, Crossan et al 1999). When people work, acquire knowledge and communicate in an organisation, they use, share and create knowledge. The creation of knowledge in an

Reference this paper as: Rantapuska, Torsti (2005) "The Adoption of new Application Development Tools by IT Professionals from the Viewpoint of Organisational Learning" *The Electronic Journal Information Systems Evaluation* Volume 8 Issue 3, pp 197-214, available online at www.ejise.com

organisation is a circular process (see Fig 2.1), in which knowledge is created in four stages of knowledge transformation: from tacit individual to explicit individual (*externalisation*), from explicit individual to explicit organisational (*combination*), from explicit organisational to tacit individual (*internalisation*) and finally from tacit individual to tacit individual knowledge (*socialisation*).

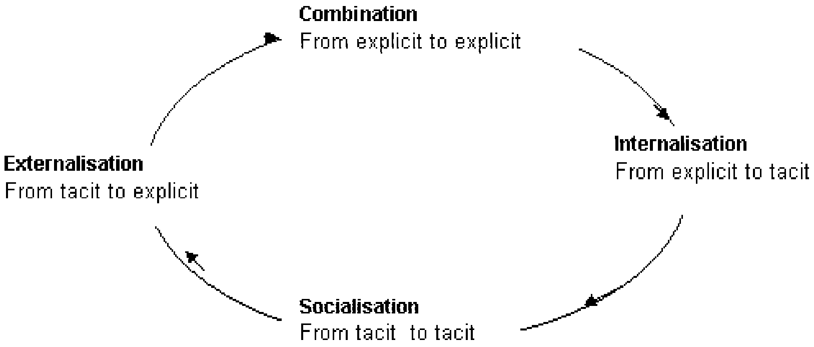

Figure 1: The organisational creation of knowledge (Nonaka 1994)

Kolb (1984) describes the process of learning in his experiential learning theory. His theory includes similar pattern of knowledge conversions as described in Figure 1, but from a personal point of view. A successful learning process includes four modes of learning (see Figure 2.) The learning begins with the learner's own (tacit) experience of *reflective observation.* He adapts the phenomena at hand by reflecting internally the (tacit) world and giving new attributes and properties to the observations. He grasps an intentional apprehension about the phenomenon. In the second phase, *abstract conceptualisation*, the learner thinks about and builds theories from these experiences into a symbolic (explicit) form. In the next stage, *active experimentation,* the learner applies and tests the concepts into practice. He or she gets an external comprehension of how these theories and rules work in practice. Finally, when applying the rules in a continuous process of *concrete experience*, the learner apprehends the world in time and space. He learns the daily routines by doing them and not having to pay special attention to each separate part.

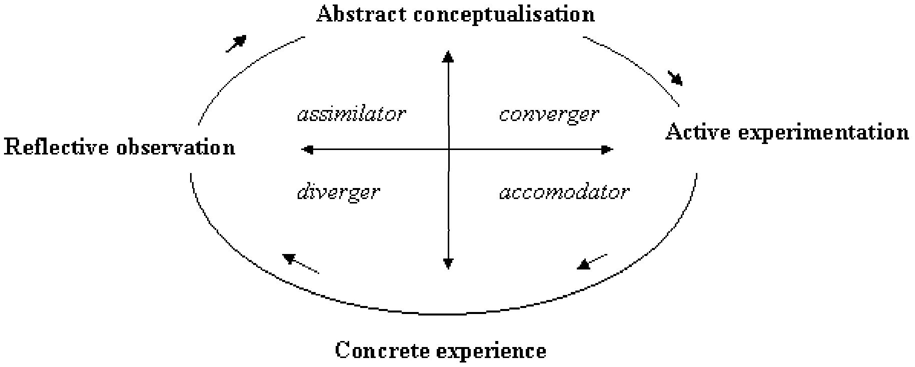

Figure 2: Experiential learning model (adapted from Kolb 1984 p. 23)

Kolb derives four different learning styles depending on which two learning modes dominate:

1. The *diverger* styles of people acquire knowledge relying on their concrete experience. They do reflective observations when thinking of the situation from different viewpoints. They are good at creating new ideas.
2. The *assimilator* styles of people acquire knowledge by reflective observation. They try to understand the meaning of the phenomena as a whole and use the abstract conceptualization mode to adapt their reflections into a theoretical framework.
3. The *converger* styles of people base their knowledge on abstract conceptualization and act by doing active experimentation in order to apply theoretical knowledge into practice.
4. The *accommodator* styles of people acquire knowledge by active experimentation. They like "learning things by doing" based on concrete experience.

Järvinen and Poikela (1997) have complemented Nonakas and Kolbs frameworks of organisational learning by adding the group level (Figure 3).

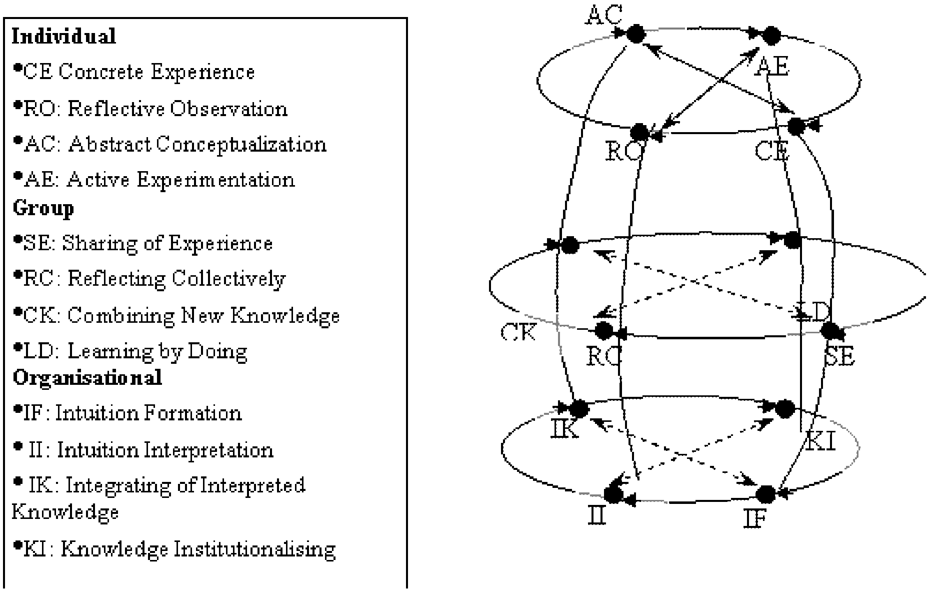
Individual
•CE Concrete Experience
•RO: Reflective Observation
•AC: Abstract Conceptualization
•AE: Active Experimentation
Group
•SE: Sharing of Experience
•RC: Reflecting Collectively
•CK: Combining New Knowledge
•LD: Learning by Doing
Organisational
•IF: Intuition Formation
• II: Intuition Interpretation
• IK: Integrating of Interpreted Knowledge
•KI: Knowledge Institutionalising
AC
AE
RO
CE
CK
RC
LD
SE
IK
KI
II
IF

Figure 3: Adapted from Järvenpää and Poikela (2001)

According to their model, the same stages of learning in an individual level may be expanded into useful activities in-group level as well. Concrete Experience in the individual level may be expanded into *Sharing of Experience*, Active Experimentation into *Learning by Doing*, Abstract Conceptualization into *Combining new Knowledge* and finally, Reflective Observation into *Reflecting Collectively*. The group level works as a link between the individual and organisational levels.

Another feature of personal learning analysed in this study is motivation (Ryan & Deci 2000). The motivation is *intrinsic* when the activity is performed because it is enjoyable per se. The motivation is *extrinsic* in cases where the activity is performed because of its usefulness for other outcomes.

When taking an organisational view to learning, we see the work as its *objective* institutional properties like sequential workflows and discrete functions described in processes and methods (Orlikowski and Robey 1991). In the *subjective* level, the work is described as a set of tacit, unstructured social actions by taking *a* look at how working people communicate, think through problems, forge alliances, and learn as a way of getting work done (Sachs 1995). The subjective and objective views are not mutually exclusive. The human actors, when exhibiting the structural properties in their work, also reproduce these properties in the actual organisational context. The subjectivist view is important when stressing creativity and flexibility, whereas the organisational view is useful when stressing the effective enforcement of organisational tasks.

2.3 Research methods

This study is an intensive case study with a *convenience sample* (Cunningham 1997) of interviews of fourteen IS-professionals who have recently started using a new application development tool. The data was gathered in 2004 from the regions of Lahti, Kotka and Helsinki (Finland). The tool adopted was Microsoft Visual Studio.NET. All interviews were recorded and transcripted on paper. The same themes and questions were used in the interviews, but following the order of the situation and in terms of their own experiences. At the end of the first interview, the informants were asked to fill in a questionnaire to test their personal learning styles (Kolb 1984, for the test used, see Virolainen and Kerola 1990) and another short questionnaire sheet to measure the usage of the functions of the tool. The informants were asked the following questions:

Adoption

1. Could you tell me about your programming background and personal preferences? How did you get started with VB.NET?
2. Was there any other VB.NET user in your workplace? How did you get support from them?
3. What do you do when you face a problem?

Usage of the Tool

1. What do you think about the improvements of VB.NET in relation to the earlier tool used?
2. Do you use the automatic code generation of VB?
3. To what extent do you use ready-made objects and controls?
4. Have the object-oriented features of VB changed the way you write programs?
5. How do you use design documents in your work? Do you have a special tool for that?
6. What kinds of methods do you use?

3. Data analysis

3.1 About the data

The working experiences of the persons varied between one and five years. Nine persons were under thirty years of age. The educational background of eleven persons was university level, from them eight had a polytechnic education and three of them had a degree from traditional university. Five of them were still studying, but working beside their studies. Four of them had studied business oriented information technology and ten had studied natural sciences. The three people having secondary level of education had graduated from a business college. All the people interviewed were male.

3.2 Data analysis

3.2.1 Motivation

The people in this data have been quite active in playing and working with computers and other technical equipment. The motivation for most (8) of the persons was based on intrinsic interest in a new tool, in this case MS.NET (see table 1).

Table 1: Motivation and original decision to adopt

	Original decision to adopt			
Motivation	Own decision	Proactive	Task assignment	Grand Total
Extrinsic			2	2
Intermediate	1	1	2	4
Intrinsic	2	2	4	8
Grand Total	3	3	8	14

Four intrinsic oriented persons started to study the tool in advance. An opinion leader or the huge amount of information on the Internet inspired them. The two extrinsically oriented persons adopted the tool as a result of an organisational decision. Four persons expressed both intrinsic motivations and extrinsic motivational factors.

The younger informants took the new tool seriously and saw that it could contribute to their careers. They were eager to attend courses and learn the tool at home as well. The less experienced did not take so much extra effort to learn it in advance. For the most experienced professionals the new tool was just one tool among the others. These people also took a more critical stance towards the new tool. Although Visual Basic was not regarded as the same language any more, it had some "*historical baggage*" in which things are done "*a bit ho hum*".

3.3 Learning and working styles

3.3.1 Learning and working styles according to the test

The personal learning style inventory exposes the profile of the person as a learner. The four axes indicate the dominance of each learning style in the process of acquiring knowledge and acting towards the object under study. The dimensions are shown as follows (Figure 4).

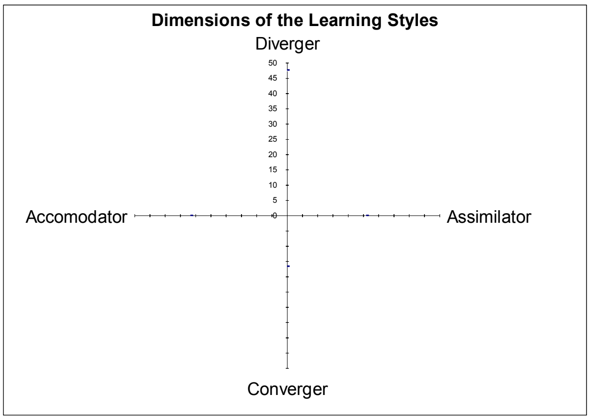

Figure 4: The dimensions of personal learning styles

The results of the measurement for each learner are shown in Figure 5. From Figure 5 it can be seen that even if there is a slight orientation towards the *diverger* and *assimilator* styles, no learning style dominates.

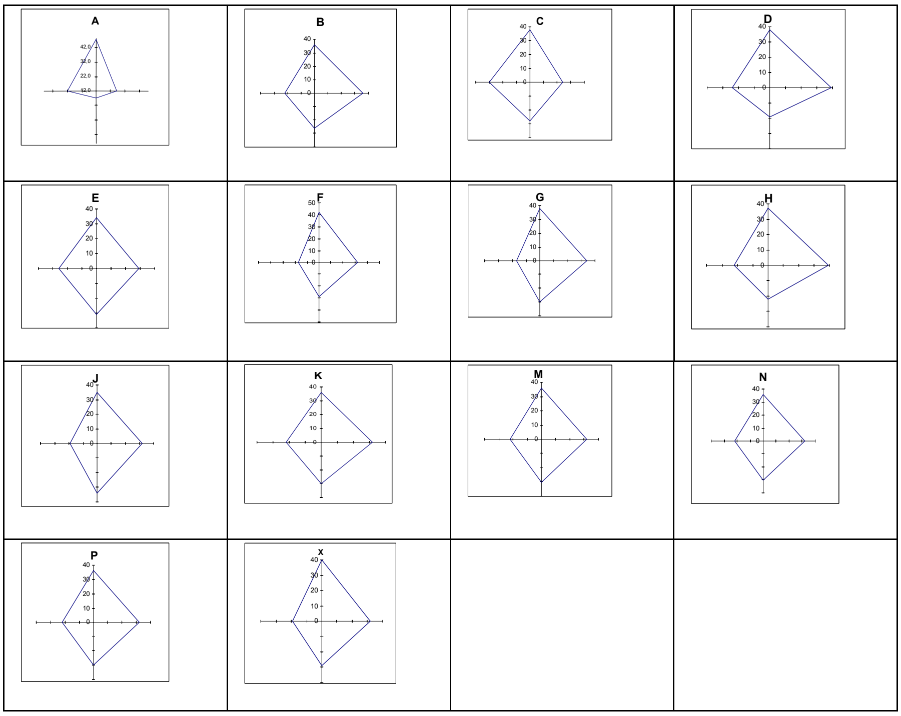

Figure 5: The personal learning styles of the IS professionals

3.3.2 *Learning and working styles according to the interviews*

The ways the interviewees describe their first experiences of programming expose quite a concrete style of learning. These stories indicate unsystematic, repeated trials such as playing, trying, and testing. When starting their professional careers, various channels become available for them to demonstrate their personal working and learning styles in practice. Table 2 lists the information channels mentioned by the informants.

Table 2: Source of information channel used when solving problems

Informant	Source of Information Channel
A	Opinion leader, help systems, internet, courses
B	Peers, opinion leader, help systems, internet, courses
C	Peers, help system, courses
D	The internet, text books, courses
E	Opinion leader, discussion group, help systems, internet
F	Help system, internet
G	Help system, internet, courses
H	Peers, help systems, internet, text books, courses
J	Peers, discussion group, help systems, internet, text
K	Peers, opinion leader, discussion groups, help systems, internet, text books courses
L	Discussion groups, help systems, internet
M	Discussion group, help systems, internet, text books, courses
N	The internet, text books
P	Peers, help systems, internet, text books, courses

The Internet and help systems seem to be used most widely. The commercial short courses are also popular. Especially young professionals mentioned them. The courses were used for the purpose of "*getting-started*" in order to get "*an idea about what one wants and what the tool is able to do*". The advanced courses were mostly used as a meeting point to get solutions to specific problems from the instructors. The actual knowledge and skills are acquired by doing the work itself. Some of them had already started to learn the tool as a Beta tester.

The traditional style of reading books is not regarded very much as a basis of learning among the informants. The situated nature of learning becomes apparent in the style how these books are used:

> *« Well, the text books are that like that you can learn the issue, but, the things go in an opposite way. You have to study blind the one that you want to study… There is an issue, you study this thing from that. Then you need it somewhere and maybe remember it in some stage. But for a problem*

> *solving these books are not very applicable. »B)*

Mostly those persons who use several sources mention the textbooks. Also the persons who used them used them as a "*reference book*" or just to learn the basic things of the programming language.

3.3.3 Communicating with peers

Six of the informants express to have used their peers in the context of knowledge acquisition. Their role still remained as a second hand source. Asking help seems to be something that should be avoided.

> *« I kinda feel like I don't necessarily want to bugger others with every single problem, you know « (G).*

The most widely mentioned sources of knowledge are the help system and the internet. Microsoft's help system MSDN is perceived to be easy to use and also very useful. The Internet and its search engine Google were mentioned by all of the informants.

3.3.4 Method of solving problems

The usage of the Internet and help systems is seen as a method of getting a solution to a problem in the form of a code example or as an answer to a specific question from a colleague in a discussion group. Not depending on the source of knowledge used, the informants usually seek a pre-digested solution to a problem at hand.

> *«'Copy-paste' is a killer, so copy and use it. I'm the kind of programmer who uses a lot of readily available code because it cuts my workload very much » (P).*

The problem solving is usually regarded as a task of seeking information and not as a task of hard thinking, experimenting or testing possible solutions. The programming itself as a problem of algorithm design was not an issue.

3.3.5 Usage of ready-made components

The usage of different features of VB.NET is shown in table 3. The average usage of each function shows the popularity of each function. The most popular one was the "Using of own objects" with 11 users, "Dataset controls" with 10 users and "Intellisense" with 10 users.

The interviews reveal the same level of function usage as in table 3 The 'intellisense' was the most common issue mentioned. The informants also talked very eagerly about the easiness of creating forms and processing databases. The object-oriented programming and XML are also widely acknowledged as new, interesting and easy-to-use features.

Table 3: The average of functions used in MS.NET

Function	Average	Number of users (used very often)
Own Objects	3,8	11
Dataset	3,6	10
Intellisense (immediate help)	3,6	10
Validation Controls	3,4	6
XML	3,4	7
Inheritance	3,3	6
InterfaceBasedProgramming	3,2	6
Treads	3,0	4
Web Services	2,8	4
Comment Generation	2,7	4
View State	2,6	3
Web Forms	2,5	2
Cache	2,4	2

To summarize the profile of personal learning and working styles, we can say that the developers, when solving problems in adopting a new tool, use more active and experimentation orientations. Their basic personal model to act seems to be searching the solution to the problem rather than trying to find the core idea or theory to use in order to solve the problem. Their goal seemed to be to get the work done and not to continue trying to find the best and most elegant solution. This is apparent in both of the sources of knowledge they use and also in how they use these sources.

3.4 The effects of the adoption

3.4.1 Changes in the style of programming

The object-oriented features of the tool seemed to be a very important issue for the developers, especially for those to whom the tool was new. The developers had some knowledge about The Object Oriented Programming (OOP) from the school, but most of them were still learning to use OOP methods in their programs. The tool itself seemed to be the main reason for that. The tool forced the developers to use the OOP and also provided immediate help with the usage of objects. For many of the developers, the VS.NET was the first time when they understood "*what the objects really are*".

3.4.2 Use of systematic methods and preparation of documentation

The use of systematic design methods and documentation seems to be a big issue among the informants. Generally, the developers prepared no design documents or documentation for the applications developed. The documentation work was restricted to writing program line comments. However, the importance of the design documents was acknowledged. The arguments for neglecting the preparation of documents were given in a way as if looking for an excuse. Usually the informants complained about the lack of time, lack of skills, or the small size of their projects.

> *Especially it's a matter of… everybody knows that it's clearly beneficial the bigger the project becomes, then its … like an application for one method only starts to bear fruit, but maybe they always think that it's like too much work and this is such a small project so it doesn't pay off to develop such an environment. Maybe this is the traditional way to look at it. It is also surely my own resistance against change » (B).*

The most experienced developer even denied the need for preparing documents altogether. According to him, the documents are useful only to the novice developers.

To summarize, the tool has some positive effects on the working methods of the developer, even if they remain few. In particular, the young developers count on the tool they use and also try to use its functions. The tool may also have positive effects on the methods of programming, but not on the use of systematic working methods in general. The developers seem to take the preparation of documents as a task that may be important for the organisation, but leaving it undone, does not indicate an essential lack of professional skills.

3.5 Organisational learning

3.5.1 Learning as a group interaction

The developers do *share experiences* (SE) to some extent, but not very often. The situations of SE are usually cases where a colleague gives a direct answer to a problem at hand. In many cases, it is also a teaching situation in which a more experienced developer shows how a particular case should be done. However, the consultation very seldom occurs from the initiative of the one who needs help. Asking for help indicates some level of helplessness.

> *« Somehow I feel like, I don't want to bother others with every problem, or like.. there has been like, when everything is new to everybody… »(G)*

The tasks of the developers were mostly individual assignments in nature and, accordingly, didn't provide opportunities to *learn by doing* in groups. The situations in which the developers *combine new knowledge (CK)* or *reflect collectively (RC)* are also few. The RC usually takes place in meetings and briefings for planning the system and agreeing on issues in common. Adoption and internalization of new tools seem to be at an individual level of IS people. They may ask for help, attend courses together, give advice, but not to the extent of working and learning in knowledge creating teams. The professional identity of IS people seems very much to be rely upon the myth of getting along on your own.

4. Conclusion and discussions

The main purpose of this study is to describe the adoption of new IT tools and how the adoption of these tools affects the individual working methods. Consequently, I described the adoption, learning and usage of new application development tools among IT-professionals. The learning styles are analysed based on the standard test as well as on analysing the interviews. The usage of tool is analysed by finding the ways that the developers acquire knowledge, how they solve problems and to what extent they use the features of the tool. Secondly, I analysed the changes in the working methods, especially in relation to the use of systematic working methods in designing, implementing and documenting of applications. Finally, the data is also analysed through the lens of organisational learning.

4.1 Key findings

As learners, the IT-professionals are quite many-sided. They express themselves to be both theoretically and practically oriented indicating only a slight orientation towards the diverger and assimilator style of learning. The working style in real working situations exposes a more practical and context-dependent nature in their style of learning. In these situations, the developers solve problems as far as they arise from the work itself but not preparing for the task in terms of ideas and work plans. This is why they usually use help systems and the Internet as their information channels, because they provide an immediate solution to the problem at hand.

The tool itself has an effect on the style of working; it guides the developers to write programs in using the object-oriented way. The same effects in the usage of systematic methods and preparing of documents did not occur.

The IS professionals are individualistic and personal in nature. When performing their organisational tasks, their primary intention is to get the work done and not to follow formal working methods too much. They define their tasks in terms of technical properties: the task is accomplished when the application works properly. They do not co-operate very much face-to-face with their colleagues, but more likely try to manage on their own. The anonymous members of discussion groups on the Internet seemed take the place of their colleagues in that role.

4.2 Discussions and earlier findings

The earlier results support the findings about the diversity of learning styles among IS professionals. The IS professionals are orientated to disciplined, but also concrete working and learning styles (Kakabadses 2002, Kolb 1984, Fowler et al 2000, Soloman 1992, Felder & Silverman 1988). However, when learning is a necessity to get the work done, the active and concrete side of learning dominates. This context-based orientation is widely acknowledged in the literature (Brown & Duguid 1991, Engeström 2001).

The low level of group interaction and orientation to technology is also recognized by other studies (Kakabadses 2002). Communication and search of knowledge seem to have moved onto the internet (McGuire & Randall 1998). Other researchers have found similar results about the individual and unsystematic nature of working methods (Conradi & Dybå 2001) too. It is becoming more flexible, improvisational and situated in nature (Gasson 1999, Orlikowski & Hofman 1997).

The effects of the tool use of the tool on the use of systematic working methods has found by other researchers too. The developers, who use the tool effectively, are also prone to use systematic methods, and vice versa (Guinan et al 1997).

Based on the results of this and related studies it seems that the three components 1) *tool use*, 2) *systematic working methods* and 3) *group interaction* work by reinforcing each other. In order to enhance successful SPI process, we should pay more attention to the adoption of the tools as well as systematic working methods. Learning to use systematic methods also requires context-free and theory-oriented training. We should also pay more attention to the co-operative working methods. There is an urgent need for the local social communication, which the communication through the Internet cannot compensate.

References

Abrahamsson P. (2002), The Role of Commitment in Software Process Improvement, Department of Information Processing Science, Academic Dissertation, University of Oulu

Brown S. I. and Duguid P. (1991), Organisational Learning and Communities of Practice: Toward a Unified View of Working, Learning, and Innovating: Organization Science, Vol. 2 No 1, February 1991, pp. 40-57,

Brown S. I. and Duguid P. (2001), Knowledge and Organisation: A Social Practice Perspective, Organization Science, Vol. 12 No. 2. March-April 2001, pp. 198-213

Conradi R. and Dybå T. (2001), An empirical study on the utility of formal routines to transfer knowledge and experience, ESEC/FSE, Vienna, Austria

Crossan M. M., Lane H. W., and White R. (1999), An Organisational Learning Framework: From Intuition to Institution, Academy of Management Review, Vol 24, No 3, pp. 522-537

Engeström Y. (2001), Expansive learning at work: Toward an activity theoretical reconceptualization, Journal of education and Work, Vol. 14., No. 1 pp. 133-189

Eraut M. (2000), Non-formal learning and tacit knowledge in professional work, British Journal of Educational Psychology, Vol. 70, pp 113-136

Felder R. and Silverman L.(1988) Learning and teaching style in engineering education. Engineering Education, 674-681, April 1988

Fowler, L., Allen M, Armarego J., and Mackenzie J. (2000) Learning style and CASE tools in software Engineering. In A. Herrman and M.M. Kulski (Eds), Flexible Futures in Tertiary Teaching. Proceedings of the 9th Annual Teaching Learning Forum, 2-4 February 2000. Perth: Curtin University of Technology. http://lsn.curtin.edu.au/tlf/tlf2000/fowler.html

Fuggetta A. (2000), Software Process: A Roadmap, The Future of Software Engineering, A. Finkelstein, Editor, ACM Press, 2000, pp. 25-34.

Gallivan M. J. (1995), Reskilling IS professionals: Individual and organisational adaptation to software process innovations, MIT Sloan School of Management, Cambridge, USA

Gasson S. (1999), A social action model of situated information systems design, The DATA BASE for Advances in Information Systems – Spring 1999, Vol. 30 No. 2

Guinan P. J.,Cooprider J. G, and Sawyer S. (1997), The effective use of automated application development tools, IBM Systems Journal Vol. 36 No 1

Humphrey W. S., Kitson D. H. And Kasse T. C. (1989), The state of software engineering practice: A preliminary report. Tec. Rept. CMU/SEI-89-TR-1, Software Engineering Institute; Garnegie Mellon University, Feb. 1989

Järvinen A., Poikela E. (2001), Modelling reflective and contextual learning at work, The Journal of Workplace Learning, September, Vol. 13, No. 7, pp. 282-289 (8)

Kakabadse N. K., Kouzmin A. and Chatham R. (2002),´IS/IT professionals' personality difference: A case of selection or predisposition? http://www.scu.edu.au/schools/socialsciences/ajbsi/papers/vol2/is_it_prof.pdf

Kellner M. I. (1991), Non-Technical issues in Software Engineering, Panel Session Overview, Software Engineering Institute, Carnegie Mellon University, 1991 IEEE

Kolb D.A. (1984), Experiential learning, Prentice-Hall, Englewood Cliffs.

Markulis P., Howard B., Howe H.. and Gutenberg J., (1998), Does it really work? An application of the inter-group interaction (igi) framework, Developments in Business Simulation and Experiential Learning, Volume 25, 1998

Nonaka, I (1994). A dynamic theory of organizational knowledge creation, Organization Science, Vol. 5 No.1, pp. 14-37

Orlikowski W.J. and Hofman J. D. (1997), An improvisational model of change management: The case of groupware technologies, Sloan Management Review, Winter 1997

Orlokowski W. J and Robey D. (1991) Information technology and the structuring of organizations, Information Systems Research Vol. 2, No. 2 pp. 143-169

Ryan, R. M. and Deci E. L. (2000), Intrinsic and Extrinsic Motivations: Classic Definition and New Directions, Contemporary Educational Psychology 25: pp. 54-60

Sachs P. (1995), Transforming work: collaboration, learning, and Design, Communications of the ACM Vol. 38. No. 9 pp. 36-44

Seng, C,V., Zannes E., and Pace R.W. (2002), The contributions of knowledge management to workplace learning, The Journal of Workplace Learning, June 2002, Vol. 14. No. 4, pp. 138-147

Senge, P.M. (1990a), The Fifth Discipline: The Art and Practice of the Learning Organization, New York, NY, Doubleday

Senge, P.M. (1990b) "The Leader's New Work: Building Learning Organisations”, Sloan Management Review, 32 (1), Fall 1990, pp. 7-23,

Soloman B. (1992) Inventory of learning styles. North California State University, USA

Sunassee N. N. and Sewry D. (2003), An Investigation of Knowledge Management Implementation Strategies Proceedings of SAICSIT 2003, Pages 24 – 36

Virolainen K., and Kerola P. (1990), Omaksumistyylien kartoitus, Learning-Style Inventory, Oulun Yliopisto/Tietojenkäsittelyopin laitos

Wang Y. and Bryant A.(2002) Process-Based Software Engineering: Building the Infrastructures, Annals of Software Engineering archive, Vol. 14 , Issue 1-4 (December 2002) pp: 9 - 37