This is an electronic reprint of the original article.
This reprint *may differ* from the original in pagination and typographic detail.

Author(s): Nahar, Nazmun; Käkölä, Timo; Huda, N.

Title: Diffusion of Software Technology Innovations in the Global Context

Year: 2002

Version:

Diffusion of Software Technology Innovations in the Global Context

Nazmun Nahar[1], Timo Käkölä[2] and Najmul Huda[3]

[1 & 2] *University of Jyväskylä, Department of Computer Science and Information Systems, P.O. Box - 35, FIN-40351, Jyväskylä, Finland*
E-mail: naznaha@cc.jyu.fi, timokk@cc.jyu.fi
[3]*Tallinn Technical University, Department of Information Processing, Raja 15, EE0026 Tallinn, Estonia*
E-mail: nhuda@pp.htv.fi

Abstract

This study examines how software businesses are acquiring new software technology innovations (STIs) in rapidly changing globalized business environment characterized by rapidly shortening software technology life cycles, changing customer demands, and intense competition. It was executed through both theoretical and empirical investigations and analyses. It describes one case company as an example of the diffusion of STI and develops a framework for the diffusion of software technology innovation. The research findings are useful for both further research and industrial settings.

Keywords: software technology, software technology innovation, software development tools and methods, software technology transfer, diffusion of innovation, software business.

1. Introduction

Software businesses are facing increasing global competition due to the liberalization of trade through GATT agreements, the building of regional and world trading blocks, and the impacts of new information and communication technologies, multimedia tools and services, improved logistics services, and electronic banking systems. In order to compete in the market place software businesses need to maintain a strong technological position in relation to their current and potential competitors.

Software businesses are under pressure for more efficient, on time, and within budget product development [10, 17, 18, 36, 37]. Customers are increasing their demands and their needs and preferences of software products and services are changing fast. Software businesses also need to be agile and flexible so that they can quickly move into new product and market segments

where competition is minimal [3]. Rapid changes in software technologies are making obsolete the existing technologies of software businesses quickly. Businesses who can not successfully acquire the new technologies, can not fulfil the customers' requirements and survive in the long run. For these reasons they are forced to acquire software technology innovations (STIs). Software technology refers to the methods, processes, and tools that are used in developing, operating, and maintaining software. According to Rogers [31], an innovation is: "the idea, practice, or object that is perceived as new by an individual or other unit of adoption".

Most organizations follow ad hoc software technology transfer (STT) processes due to the lack of a systematic STT framework. Transferring software technology through ad hoc processes is relatively slow and costly and encounters many unexpected obstacles [34]. Due to a variety of reasons the STT results in ineffective use (or even nonuse) of software technology [10, 17, 18, 36, 37]. Furthermore, STT is a complex endeavor due to the number and nature of activities and tasks involved and the risks, uncertainties, and costs faced.

Research is necessary to identify how software businesses could implement STT more effectively. Our literature review suggests that no framework has been developed to guide managers in implementing STT in the context of rapidly changing software technology and globalized business. This paper develops a STT process framework to start filling this gap in knowledge.

The research question addressed in this study is "how can software businesses acquire software technology effectively and efficiently?" This question has been examined by conducting theoretical research in the areas of software technology, STI, STT, diffusion of innovation theory, and empirical research through a multiple focus group research method.

This paper proceeds as follows. A literature review is carried out in Section 2. Section 3 presents a framework for the analysis of STT and describes the characteristics of

an innovation as well as the characteristics of a company. The research method applied in this study is described in Section 4. Section 5 describes the STT and diffusion process of KONE. The factors that contribute to the success of STT are also highlighted in this section. Section 6 develops a framework for the diffusion of STI which consists of twelve phases. Finally, conclusions are discussed in Section 7.

2. Software technology and software technology transfer

2.1. Software technology

Software technology refers to the software development tools, methods, knowledge, and skills that aid in performing the software development tasks [17, 34]. CASE tools, software development methods and processes are common components of the software technology.

2.2. Software technology transfer

STT refers to the transfer/delivery of a software technology package (hereafter "package") by a technology producer/supplier (hereafter "supplier") to a technology consumer/recipient (hereafter "recipient") who uses it in the creation and production of software products [17, 34]. The package includes software technology, documentation/procedures, training and technical support. For example, when a supplier delivers CASE tools to a recipient only to be used by the recipient for developing application software and does not provide the recipient with training and support necessary for producing the CASE tools, this process is not considered to be STT. It is considered to be selling or diffusion of software products.

STT (i.e., the diffusion of software technology innovation) and software product transfer processes (i.e., diffusion of software product such as an enterprise resource planning system) are two completely different concepts.

Orlikowski [24] analyzed in great detail a software business that faced many challenges typical in software industry. Productivity in customized application development business was low, personnel turnover was high, and software developers were highly skilled and expensive experts that considered themselves more like individual artists than members of a cohesive product line and worked accordingly. Company management introduced a CASE tool in the organization to enforce standardized development methods, reduce the skill and respective salary requirements of software developers, and measure and control the productivity of individual developers. The management was successful in increasing productivity and reducing organizational dependency on expensive individuals by introducing the tool. The developers did, however, manage to negotiate changes in the tool, thus reducing the degree of managerial control mediated by the tool and methods largely because they had excellent software related expertise.

The diffusion of CASE tools has some characteristics similar to those of STT. For example, both the providers and the end users of CASE tools are software professionals and have extensive and specialized software related knowledge. The diffusion also typically involves a significant organizational change within the recipient in terms of business processes, technological infrastructure, culture, required competencies, and many other factors. For this reason, Orlikowski's study is interesting but even her study has limitations from the viewpoint of STT. The study describes the diffusion of a software product and does not deal with STT at all. Moreover, the supplier and the recipient were two units of a large software company and operated in the same cultural and geographical context. STT also involves much more extensive organizational change than the diffusion of standard software products.

To make the differences between the concepts of the diffusion of software products and STT clear, another example can be given. The successful adoption of an enterprise resource planning (ERP) system product necessitates special training and intensive technical support from the supplier as well as a very competent project manager for the implementation of the project. The magnitude and complexity of organizational change involved in ERP implementation are typically in the same range as those of STT. However, the organizational implementation of ERP or any other software product can not be regarded to be STT because the recipient receives training and technical support from the supplier only to the extent required for using the product. In this case, the recipient will not have the capability or know-how to produce similar software products and services it has acquired.

Although there is a great need for a framework of STT, we have found very little research on how STT is executed in practical situations and which activities are performed during this process. Marciniak [17] researched "What is software technology transfer and how is it facilitated?" However, there is very little discussion in the article about the real STT process. Furthermore, Marciniak [17] examined "the diffusion and adoption of software as a way to understand STT". As our examples have illustrated, STT is very different from diffusion of software products.

Great amount of effort, financial resources, technical expertise, and time are needed in STT in order to transfer required knowledge and skills concerning product and

process technologies, operations, and maintenance so that the recipient can produce the same quality software products and services as the supplier. Eventually the recipient can even take the supplier's role and, for example, start STT to third parties.

The characteristics of the technology transfer projects differ between recipients in developed and developing countries. Companies in developed countries usually need legally protected technologies (i.e., patented product or process technology). As such, technology transfer projects in developed countries are not very complex [21].

In contrast, recipients in developing countries require product and process technologies as well as a broader range of skills, expertise, and procedures related to operations, problem solving, and management. Thus, the technological needs of a company in a developing country are composite in nature and may go beyond the various stages of project preparation and implementation. These needs include operations, maintenance, management, and marketing. Therefore, technology transfer projects in developing countries are complex [21].

Recipients and suppliers usually co-operate actively during STT exchanging requirements, funds, software development tools, and services, and learning from each other through mutual feedback. STT is a continuous process. Technology is dynamic, new technology making the older ones obsolete. A supplier will usually transfer new technology to the receiver according to pre-negotiated agreements.

We thus conclude that the diffusion of software products is different from software technology transfer and these processes should not be confused. Future research could, however, investigate the diffusion of certain highly complex system products especially in contexts where users are highly competent in software related issues. Insights from such contexts might be useful for understanding STT and vice versa.

3. A framework for the analysis of software technology transfer

Diffusion of innovation (DOI) theory helps to understand the diffusion of innovations. The main objective of STT is to facilitate the transfer of software engineering methodology and reusable software components that will help the recipient to develop software products and services. In STT, the recipient acquires a new software technology. This technology is one kind of innovation. Through STT, this technology is delivered to the recipient. In this way, STT can be viewed as the diffusion of software engineering innovation. Therefore, the DOI theory factor approach is suitable for examining the current research problem.

Rogers [30] introduced DOI theory in his book "Diffusion of Innovations". According to Rogers [31],

diffusion is "the process by which an innovation is communicated through certain channels over time among the members of a social system".

Researchers are increasingly studying innovation diffusion [1] and IT diffusion [7, 16, 28]. DOI theory has been widely applied to the study of IT innovations to gain insights into their implementation [26]. Characteristics of innovations and companies play crucial roles in the implementation of innovations [21]. A more extended version of the diffusion of innovation framework can be found in [21].

3.1. Characteristics of an innovation

Based on the research work of Rogers [30-32], it is apparent that an innovation is more likely to succeed if it includes the following characteristics: relative advantage, compatibility, complexity, trialability, and observability. The research results concerning the implementation of innovations [13] and IT innovations [25] are consistent with the above findings of Rogers.

Researchers have found out additional innovation characteristics that influence implementations of IT innovations. Hoffer and Alexander [8] discovered that technology functionality, performance and efficiency played an important role in the successful implementation of IT innovations. Russo and Kumar [33] found ease of use, task productivity and task quality were important for the success of implementation of IT innovations.

3.2. Characteristics of a company

The characteristics of a company have long been associated with its capacity for successful implementation of innovation [6]. Based on the literature review, the key characteristics identified include: need for the innovation [29], availability of resources (i.e., material and financial resources, knowledge and skills, time, experience) [4], management of innovation implementation [16, 22], risk management [21], management support [13], leadership, champions, motivation, participation, and training [4].

General diffusion theory can not properly explain diffusions of many types of innovations such as complex and networked IT solutions [15]. Researchers are extending DOI theory [27] and integrating it with other theories [5] in order to explore the diffusion of IT innovation.

4. Research methodology

This paper draws upon and reports parts of a long-term research project that has investigated the diffusion of software products in globalizing firms and the diffusion of software technology in macro environment for many years. The project has also conducted research on

diffusion of software technology in organizations. The research question, the research framework presented in Section 3, the literature review in the fields of software technology and STT, and the questionnaire guide developed in [21] acted as a basis of conducting the field study.

This study examines "how" and "what" type of a research question. A research method is needed that is suitable for answering such types of questions. A focus group method is suitable for "what", "how" and "why" types of questions [11, 20]. A focus group is "a carefully planned discussion designed to obtain perceptions on a defined area of interest in a permissive, non-threatening environment. Group members influence each other by responding to ideas and comments in the discussion" [11]. In the focus group method, group interview and discussion produce data rich in detail. Participants' views and experiences are expressed through interaction and open discussion [35]. Social scientists are increasingly employing focus group method in their research [20]. For these reasons, a multiple focus group research method was selected to conduct this research.

By following the guidelines suggested by the methodologists [11, 20] and drawing on the literature review on STT and the diffusion of innovation theory, we developed the focus group questionnaire guide. Then the guide was reviewed by researchers and tested in a group discussion.

Three focus group discussions were conducted with knowledgeable informants in three Finnish suppliers investigated during 1998 to 2000. These suppliers are involved in production of software for: elevator and escalator production and maintenance, energy production management, and paper factory automation. They have successfully transferred software technologies to several software technology recipients around the world. The recipients located in India, Thailand and Hungary were interviewed by using e-mails, conferencing technologies, and telephones. The focus group participants and interviewees were IT managers, business or general managers, and other executives who were either directly involved in the STT projects or were connected to the projects. Each focus group discussion lasted 4 to 5 hours with some intervals.

To investigate the research question, the informants were asked the following sub-questions:

- How were software technologies transferred to the recipients?
- What were the phases in the STT process?
- Which tasks were performed in each phase?
- What were the factors that were critical for the success of STT? How did these factors enhance success in STT?

Interviews focused on each of the identified phases of STT. In order to enhance interaction between focus group participants and facilitate open and in-depth discussion of all participants' views and experiences, strict guidelines suggested by focus group methodologists were followed during each focus group session.

According to the suggestions of research methodologists [12, 19], several activities were performed to conduct the data analysis for each focus group discussion. We took field notes during the discussions. We also recorded the discussions, listened to the tapes, and wrote down the important ideas that were related to the research questions. This study followed the following steps for qualitative data analysis: data reduction, data display, individual focus group analysis, comparative focus group analysis and conclusion drawing.

5. Software technology transfer and diffusion process of KONE

This section deals with the investigation and analysis of factors contributing to success and factors creating barriers to STT. We describe a focus group analysis for one investigated company, a Finnish multinational called KONE. KONE is one of the world leaders in manufacturing, installation, maintenance, and modernization of elevators and escalators. We highlight the diffusion process of software development tools and methods into an internal environment of KONE's one subsidiary in India. The analysis has been performed on the basis of the research framework. We focus on those replies of the focus group participants that were supported by all other participants during in-depth discussions.

5.1. Case analysis and description

The recipient is KONE Elevator India Limited (KEI). KEI develops software for the maintenance of elevators and escalators. Its customers are located around the world. It also develops manufacturing, logistics, accounting, and other types software to be used by the subsidiaries of KONE worldwide. Twenty software programmers and analysts were working in this subsidiary. Two of them were development leaders. KEI also taps outside software professionals as contract software developers on a regular basis. An expert, who was an experienced, qualified chartered accountant and software specialist controlled the overall direction and major software development projects.

There was limited software development activities in KEI in 1980s. In the beginning of 1990, KONE decided to expand software production significantly in KEI. The reasons included:

- The need of high performing software for logistics, manufacturing, and maintenance of elevators.

- Availability of the low cost software professionals in India.
- Incentives offered by the Indian government to make software for foreign markets.

A STT project was started to enable the expansion. The following statements illustrate the complexity inherent in STT. An IT manager from KONE headquarters in Helsinki stated:

"In order to transfer software technology and increase software production in KEI, India, we performed various activities. The major ones include external and internal environment analyses, identification of needs, commitment development of the organization, success factors, risk factors, planning for the STT, etc."

The project manager of the STT project mentioned:

"We performed various tasks to transfer software technology to KEI. For example, we expanded its computing capacity, recruited highly educated software professionals, selected high-performing software development tools, delivered software development tools and methods, provided training, developed the required software, and evaluated the software technology transfer performance."

The technology package acquired by KEI included the following elements:
- Management of software projects - identifying the project process, allocating resources, managing the subproject, controlling, reporting, etc.
- Software quality management - training on quality methodology to improve the quality and meet the required quality standard.
- Modern software development tools and relevant training so that the tools can be used in the software projects.
- Creating documentation - how to develop software manuals according to customer needs and corporate standards.

KONE sent two trainers to India to train the KEI software professionals intensively and adequately. KONE also provided them with training tools, materials and methods. KEI employees themselves did lots of training. Software development tool vendors also provided training and technical support to KEI software professionals.

KEI's employees also traveled to KONE's software development centers to be trained. KONE transferred software technology successfully by providing documents, training Indian software professionals in the centers, and allowing them to observe KONE's software production facilities and processes. KONE's culture was open enough to let the professionals come inside the company and share knowledge with colleagues in Finland

and other Western countries. There were no major restrictions. On the other hand, the professionals were capable enough to absorb and take with them the relevant knowledge. The professionals were well educated and also highly motivated to learn more for several reasons. For example, due to the professionals' improved competence after having been trained, other companies offered many of them higher salaries and several professionals left KEI over the years. The STT project also encountered barriers due to complexities of importing computer hardware required to utilize software development tools and produce software in KEI. There were strict import restrictions and very high import taxes for computers, associated spare-parts, and peripherals. Unreliable electricity supply also caused problems. Over the years KEI has been able to cope with these and other problems and develop software for logistics and manufacturing and maintenance of elevators successfully.

KONE evaluated KEI's performance when the STT project had been completed. It found that the software projects of KEI were frequently falling behind the schedule because KEI professionals were not competent enough in managing complex software development projects. It partially solved the problem by recruiting experienced personnel and providing additional training.

KONE is highly advanced in using IT to support STT and collaborative multi-site, multi-partner software development projects across its software development centers. For example, it is using video conferencing for sharing software development knowledge and skills among its professionals in the USA, Finland, Australia, and other countries. However, it is not able to use video conferencing for India due to low bandwidth and reliability and high communication expenses of Indian telecommunication networks. Partly because of this, KEI cannot participate fully in such projects and KEI professionals cannot develop their competencies optimally to coordinate, manage and control such projects very effectively.

KONE is delivering more powerful software development tools, methods, and related training to KEI on a regular basis. However, KEI encounters some difficulties to acquire all the tools and training that it needs due to high financial requirements. Currently, KONE uses CD-ROM-based interactive multimedia training, feedback through e-mail communication, and teleconferencing technologies along with commonly used training tools and methods to help the recipient's employees adopt and use the new software technology effectively and efficiently. KEI is acquiring advanced software technologies continuously from IT vendors and KONE software development centers.

5.2. Factors that contributed to the success of software technology transfer

The focus group data analysis revealed that the following factors played key roles in making the STT endeavor successful:

- *With some adaptations the recipient implemented proven software development tools and methods that had already been implemented in various countries.* They accelerated STT and improved performance of the recipient, thus motivating the professionals to use them. As a result, *software products and services* could be developed that are of high quality, improve productivity significantly, are consistent with software technology trends, and can be competitively priced and fulfil the requirements of the customers and the customized needs of the end-users.
- *Availability of adequate financial resources* [9]*, technically and managerially skilled software professionals, technical support staff, hardware, software, and manuals* helped implement the STT project efficiently.
- *Long experience in implementing the same standard tools and methods* helped KONE implement the STT project efficiently in India.
- The project leader of KONE supervising the STT project in KEI was competent and had excellent *leadership* skills.
- *Top management support* (c.f., [14]) was provided effectively by KONE's top management to facilitate the implementation of STT.
- *Providing adequate training and delivering proper training technologies and materials* helped Indian professionals use the development tools and methods (c.f., [2, 14]).

6. A framework for the diffusion of software technology innovation

The ad hoc STT project implementation is slow, costly, and ineffective. A review of current literature [10, 17, 18, 34, 36, 37], a cross focus group analysis of three group discussions in three companies, and our long experience in this area suggest that effective STT in the context of recipients should be considered as a process of twelve interlinked activities. Some of them are executed in a chronological order, while others occur concurrently. The STT process framework of this research is a further developed version of our earlier research [23].

On several occasions a pilot project of the transfer in a part of the organization is performed to test software technology, identify various problems and devise corrective actions. All activities of the STT process are not performed in pilot implementations.

1. External environment/factor analysis. Our research findings suggest that in order to make STT and utilization of software technology effective, software businesses should analyze the external environment/factors very carefully since they are playing an increasingly dominant role in the diffusion of software technology. In this phase, the technology recipient examines and analyses its customers' requirements, technology suppliers' capabilities, current and future market environment, software technology trends, government regulations, and other factors. Software businesses acquire new software technologies in the contexts of their technology strategies to fulfil customers' new requirements. Customers' demands and possible reactions to new technology need to be examined. Customers' inputs regarding the selection of new software technology will help to implement the technology successfully and use it effectively in the long run.

Software technology supplier plays an important role in the success of the technology transfer. Technology suppliers' capability of: adapting the software technology to suit the recipient's requirements; providing training, technical support and documentation; and introducing advanced versions of the software technology in the future need to be examined.

Software businesses also need to examine current and future needs of the software market as a whole and the major growth segments of the market. The recipient should investigate which technologies the competitors are using or planning to use. Major software technology trends with respect to the technology of interest should be evaluated. Also government rules and regulations have to be taken into account.

2. Internal environment/factor analysis. Internal environment analysis involves a critical assessment of human (e.g., technical and management staffs) and financial resources, time, technology strategy, STT experience, compatibility of existing computing hardware and software with the new technology, computing capacity, training facilities, organizational culture, management, and technical support of the recipient. The recipient's existing hardware and software are thoroughly examined to figure out whether they are compatible with the new software technology and whether they will support the use of the new technology.

3. Deciding preliminary objectives for STT. On the basis of the external and internal factor analyses, the STT decision is taken and the preliminary short term and long term objectives for STT are defined. The short-term objectives are the objectives of each phase and the long-term objectives are expected benefits from STT such as improving software development productivity. The short-

term objectives should be consistent with long term objectives.

4. Developing and sustaining organizational commitment. Facilitating STT is a complex process and requires contributions from several sources. Therefore, commitment of all stakeholders is needed. The tasks include developing and implementing an influence strategy to make all the important personnel committed to STT, getting loyalty of the organization and especially its top management to STT, ensuring preliminary allocation of resources to perform STT tasks, and keeping everybody committed to STT decision.

5. Defining technology requirements. The required technology functions and features are identified by conducting in-depth interviewing of the end-users, analyzing the existing computing facilities, and considering the STT objectives.

6. Screening and selecting a suitable technology. The recipient identifies few alternative technologies, which address its technology requirements. These alternatives are examined in-depth: how they meet the technology requirements and fit the business and technology strategy as well as organizational culture and current practices. The vendor's reputation and performance are also examined. Finally, the most appropriate technology from a suitable vendor is selected.

7. Identifying the factors that contribute to the success of STT. STT is very complex. Making it successful requires examination and identification of those factors that play important roles to the success of STT process. Common factors related to software technology, the supplier, and the recipient enhance success in STT. They include:
- Availability of appropriate and adequate financial, human, and computing resources.
- Availability of skilled software professionals.
- Competitively priced, high quality, and customized software development tools from the vendors.
- Competent IT project leader.
- Proactive management support.
- Effective training, training materials, and technical support from the supplier and IT vendors.

8. Identifying problems and their solutions. There are common problems that occur during STT in a global context. These include but are not limited to high personnel turnover, lack of reliable and high-bandwidth telephone and communication lines, and cultural differences. Especial attention for identifying the problems and formulating their solutions should be given for the transfer of complex software technology. When

recipients are located in developing countries, transfer and diffusion of software technology may encounter great difficulties. This is partly due to the unique nature of each country. The recipient and the supplier should identify the problems and mechanisms to prevent them, agree on these prevention mechanisms, and commit to using them.

9. Planning the implementation of STT. A project leader is identified and an implementation plan is formulated which includes (but is not limited to):
- Refining the preliminary objective and fixing the specific objectives.
- Developing a high-level plan describing major tasks.
- Deriving a lower level plan from it.
- Showing sub-tasks in more detail.
- Including a timetable and sequence for transfer activities.
- Identifying the resource requirements.
- Staffing the project and assigning responsibilities.
- Defining training and support tasks and data collection requirements.

10. Implementing the software technology transfer according to the plan. A pilot project is initiated if the software technology transfer project is large, all tasks of the implementation plan are performed, strategies are implemented to deal with the resistance to change, and continual support and adequate training are provided to end-users.

11. Monitoring the implementation. Implementation data are collected and the progress of the implementation is monitored and evaluated in relation to the schedule defined and cost and performance needs set in the implementation plan. The progress status of the transfer is communicated and informed to all the participants and interested parties, and a close cooperation between the supplier and the recipient is maintained in order to overcome problems.

12. Reviewing the outcomes of the transfer. The performance of the technology is examined, the positive and negative outcomes of the transfer are studied, recommendations for the negative outcomes are formulated, and all analyses are documented and stored in a knowledge management system supporting organizational learning and sharing of knowledge.

7. Conclusions

This study was executed through both theoretical and empirical research. We presented a case, which exhibits the diffusion of STI in a practical setting in an international environment, the various factors that

contributed to the success of the diffusion process in this company, and the various factors that created barriers to the diffusion of STI. This study also introduces a STT framework describing how software businesses can implement STT effectively. This framework can systematically guide businesses in acquiring software technology innovations.

This paper contributes in terms of theory and practice. At the theoretical level, it provides the framework composed of twelve interlinked activities that can be used as a basis for further research. Among these twelve activities, "external environment/factor analysis" is playing an increasingly dominant role for the effective diffusion of software technology, whereas the existing literature has not fully recognized these important issues. At the practical level, the framework can guide managers in acquiring STIs.

We are continuing the investigation of software technology diffusion in software businesses in different parts of the world because further research will be helpful in refining the STT framework and each of its twelve phases. It is necessary to examine how macro factors contribute to the success of diffusion of software technology in organizations. Further research is also necessary to discover how software technology innovations could be transferred rapidly from one software technology development center to sites in low cost software producing countries by using modern information technology in order to execute large and complex software development projects collaboratively.

When the first major STT project between a supplier and a recipient is initiated, one of the main purposes of the project typically is to *create* a software product development and delivery organization with effective enough processes and an adequate technological infrastructure. Software process improvement (SPI) is not the major issue at this stage because systematically designed processes and development infrastructures to improve do not yet exist. This is especially evident when recipients are located in developing countries. However, when the first project has been completed successfully and an ongoing STT has been established, STT must aim at improving the processes, product architectures, and enabling infrastructures of the recipient. Linking STT and SPI has been beyond the scope of this research, but we intend to analyze our research findings with respect to the SPI literature in the future.

The use of pre-defined classes, design patterns, and other components or plug-ins that software professionals can adapt and use to speed up product development significantly have been outside the scope of this research. Yet, we believe that components are at least as important parts of the technology package as methods and production tools. Their use needs to be investigated in-depth. For example, components and their innovative integration in complex system products can have far reaching implications on the technology strategies and road-mapping processes of software businesses in the initial stages of product development. These implications are interesting in the context of STT because this level of STT is likely to help the recipients move from routine developers toward strategic product and service innovators more quickly than STT that does not exploit components.

8. References

[1] P. Attewell, "Technology diffusion and organizational learning: The case of business computing", *Organization Science*, 3(1), 1992, pp. 1-19.

[2] P.Y.K. Chau, and K.Y. Tam, "Factors affecting the adoption of open systems: An exploratory study", *MIS Quarterly,* 21(1), 1997, pp. 1-24.

[3] Cusumano, M.A. and Yoffie, D.B. *Competing on Internet Time: Lessons from Netscape and Its Battle with Microsoft*, The Free Press, New York, 1998.

[4] Ely, D.P. *The diffusion and implementation of educational technology in developing nations: Cross cultural comparisons of Indonesia, Chile and Peru,* ERIC document reproduction service, ED 331 469, 1990.

[5] Fichman, R.G. The diffusion and assimilation of information technology innovations, in *Framing the Domains of IT Management Research: Glimpsing the Future through the Past*, R. Zmud, Ed. Pinnaflex Educational Resources, Inc., 2000.

[6] Griffiths, J.M. Ronald, G.H. Ellen, A.S. and Pat, C. *Diffusion of innovations in library and information science,* Final Report, Rockville, Md: King Research, 1986.

[7] V. Gurbaxani, "Diffusion in computing networks: The case of BITNET", *Communications of the ACM*, 33(12), pp. 65-75, 1990.

[8] J.A. Hoffer, and M.B. Alexander, "The diffusion of database machines", *DATABASE*, 23(2), pp. 13-19, 1992.

[9] Holloway, R. Diffusion and adoption of educational technology: A critique of research design, in *Handbook of Research for Educational Communications and Technology,* D. Jonassen (Ed.), New York: Macmillan, pp. 1107-1133. 1996.

[10] T.D. Korson, and V.K. Vaishnavi, "Managing emerging software technologies: A technology transfer framework", *Communications of the ACM*, 1992.

[11] Krueger, R.A. *Focus groups: A practical guide for applied research,* London: Sage, 1988.

[12] Krueger, R.A. *Analyzing and reporting focus group results*, London: Sage, 1997.

[13] D. Leonard-Barton, "Implementation characteristics of organizational innovations: Limits and opportunities for managerial strategies", *Communication Research,* 15(5), pp. 603-631, 1988.

[14] K. Lyytinen, and R. Hirscheim, "Information systems failure - a survey and classification of the empirical literature", *Oxford Surveys in Information Technology*, 4, Oxford University Press, pp. 257-309, 1987.

[15] K. Lyytinen, and J. Damsgaard, "What's wrong with the diffusion of innovation theory: The case of a complex and networked technology", Institute for Electronic Systems, Department of Computer Science, Aalborg University, Working paper R-98-5010, 1998.

[16] Lyytinen, K. Nahar, N. and Huda, N. IT Supported Complex Technology Transfer in High-tech Sector, in D.F. Kocaoglu, T.R. Anderson, D. Z. Milosevic, T.U. Daim, K. Niwa, T.R. Gulledge, C. Kim and H. Tschirky (Eds.), *Technology Management in the Knowledge Era,* pp. 431-438, IEEE and PICMET, Oregon, USA, 2001.

[17] Marciniak, J. Software technology transfer, in *Encyclopedia of Software Engineering*, John Wiley and Sons, New York, pp. 1323-1327, 1994.

[18] L. Mathiassen, and C. Sorensen, "Managing CASE introduction: Beyond software process maturity", *ACM*, 1994.

[19] Miles, M.B. and Huberman, A.M. *Qualitative data analysis: An expanded sourcebook,* 2nd edition, Thousand Oaks, Calif: Sage, 1994.

[20] Morgan, D.L. *Focus groups as qualitative research*, 2nd edition, London: Sage, 1997.

[21] Nahar, N. *Information technology supported technology transfer process: A multi-site case study of high-tech enterprises*, University of Jyväskylä, Jyväskylä Studies in Computing - 9, Jyväskylä University Printing House, Jyväskylä and ER-Paino Ky, Jyväskylä, Finland, 2001.

[22] Nahar, N. IT-enabled effective and efficient international technology transfer for SMEs, in *Evolution and Challenges in System Development*, J. Zupancic, W. Wojtkowski, W. G. Wojtkowski and S. Wrycza, Eds. Kluwer Academic/Plenum Publishers, pp. 85-98, New York, USA, 1999.

[23] Nahar, N. Käkölä, T. and Huda, N. Diffusion of Software Engineering Innovation in the Global Context, *Proceedings of the 24th Information Systems Research Seminar in Scandinavia (IRIS 24),* S. Bjornestad, R.E. Moe, A.I. Morch and A.L. Opdahl (Eds.), Dept of Information Science, University of Bergen, Vol. II, pp. 655-664, 2001, Ulvik, Norway.

[24] W.J. Orlikowski, "The Duality of Technology: Rethinking the Concept of Technology in Organizations", *Organization Science*, 3(3), pp. 398-427, 1992.

[25] G. Premkumar, and M. Potter, "Adoption of computer aided software engineering (CASE) technology: An innovation adoption perspective", *Database, Special Issue of Technological Diffusion of Innovations*, 26 (2&3), 1995.

[26] M.B. Prescott, "Diffusion of innovation theory: Borrowings, extensions and modifications from IT researchers", *Database, Special Issue of Technological Diffusion of Innovations*, 26(2&3), 1995.

[27] A. Rai, and G.S. Howard, "Propagating CASE usage for software development: An empirical investigation of key organizational correlates", *OMEGA: The International Journal of Management Science,* 22(2), pp. 133-247, 1994.

[28] Ramiller, N.C. and Swanson, E.B. Towards an institutional view of information technology diffusion, transfer and implementation, in *Diffusion, Transfer and Implementation of Information Technology*, L. Levine (Eds.), Elsevier Science BV, Amsterdam, pp. 353-355, 1994.

[29] Robinson, R.D. *The international transfer of technology: Theory, issues and practice*, Ballinger Publishing Company, Cambridge, MA, 1988.

[30] Rogers, E.M. *Diffusion of innovations,* The Free Press, New York, 1962.

[31] Rogers, E.M. *Diffusion of innovations,* The Free Press, New York, 1983.

[32] Rogers, E.M. *Diffusion of innovations,* 4th edition, New York, Free Press, 1995.

[33] Russo, N.L and Kumar, K. Studying the impact of information technology in innovations in organizations: The influence of individual characteristics, innovation characteristics and the innovation introduction process, in *The Impact of Computer Supported Technologies on Information Systems Development*, K. E. Kendall, et al. (Eds), Elsevier Science Publishers B.V., 1992.

[34] Software Productivity Consortium, *Using new technologies: A technology transfer guidebook,* SPC-92046-CMC, Version 02.00.08, December 1993.

[35] D.W. Stewart, and P.N. Shamdasani, "Focus groups: Theory and practice", *Applied Social Research Methods Series*, 20, California: Sage Publications, Inc., 1990.

[36] M. Sumner, "Factors influencing the success of computer-assisted software engineering", *ACM*, 1995.

[37] J.L. Wynekoop, and J.A Senn, "CASE implementation: The importance of multiple perspectives", *ACM*, 1992.