

INFORMATION SYSTEMS DEVELOPMENT METHODOLOGIES IN PRACTICE

Abstract

This paper presents a case study, which investigates the application of information system development methods. It examines which processes and factors have an effect on the choice and utilization of development methods and techniques. The study has been conducted on the basis of three development projects in a large Danish software company. The empirical data comprises twelve semi-structured interviews with developers, project managers and representatives of the organizational unit responsible for ISD methodology introduction. Five main categories of themes, which have an effect on the use of ISD methodologies, are identified. These include 1) the extent to which a specific methodology is universally applicable, 2) the nature of the introduction process for a certain method, 3) the degree of the developer's experience with development, 4) the need for security of and between the parties involved, and 5) the developer's wish for co-determination with regard to the development process. The results of our study confirm existing research in the field and offer – based on a deeper understanding of the mechanisms which influence method utilization provided by the empirical data thoroughly analysed with the help of our Grounded theory approach - some new insights, especially with regard to the interplay and relationship of the different, identified themes. The study provides further evidence for the misconception of universally applicable methods and extends the understanding of the systems development shift towards incremental methods and towards situated use of methods. We confirm that the development context's volatility and a project's apparent uniqueness are reasons for this shift.

Keywords: IS development, IS development methodologies, IS development methods and tools, Systems development techniques, Introduction and adoption

1. INTRODUCTION

The purpose of this article is to contribute to the ongoing discussion about the practical use of information systems development methodologies. There appears to be many anecdotes around this topic and less empirically grounded documentation on the actual situated use of development methods in modern systems development (Nandhakumar et. al. 1999).

The arguments for adopting and using IS development methodologies are many - Avison et. al. mention e.g.: Accurately recording of the requirements, Making it possible to monitor the progress of the development, Providing a system within an appropriate time limit and at an acceptable cost, Securing well documented systems easy to maintain, Making it possible to identify changes as early as possible, and making it possible to deliver a system which is liked by the relevant parties (Avison et. al. 1995).

Fitzgerald extends this list to also include: A framework for the use of techniques and resources at the right time in the development process, The possibility for developers to specialize and thus to differentiate the remuneration, and also the possible standardization of the process and thereby facilitate the interchangeability of developers among projects (Fitzgerald 1998a).

In spite of these arguments it is reported from various studies, that the methodologies are often not used as intended and that the systems developers in projects are either questioning the purpose of the methods, techniques etc. specified in the methodologies or needing something else in their development work. One continuously reappearing issue is that the methods are used in a very pragmatic way and therefore the methods' underlying philosophy has no wider implication for the development situation (Bansler et. al. 1993), (Fitzgerald 1998a), (Madsen et. al. 2002), (Curtis et al. 1988).

We provide an empirically grounded study of the practical use of development methods in three projects within a large Danish software development company based upon twelve qualitative semi-structured interviews.

This article is structured as follows: Section 2 presents the literature findings that helped us understand and conceptualize this study, Section 3 describes our research method. In section 4 the case company is shortly introduced, and in section 5 our main results from the analysis are presented. Section 6 contains a discussion of these results and finally section 7 concludes on our study.

2. BACKGROUND

Research on the subject suggests there is a disparity between the way methods are formally described and the way in which methods are customized for use in practice.

Truex et al. argue that the basic assumptions underlying the concept of systems development methodologies must be addressed. They question the assumptions that information system development is a manageable, linear, universally applicable and rational process. The authors propose an alternative set of amethodical assumptions, and suggest that both views must be kept in mind when engaging in the development of information systems (Truex et al. 2000).

Stolterman supports the statement that the assumption underlying the perceived need for system development methodologies is that a developers' basic approach to developing systems is irrational, and that this irrationality must be addressed to let the process follow the ideal of rationality (Stolterman 1992).

Robey et. al. proposes a political view on systems development methodologies and describe how the activities in the traditional system development life cycle, which include systems analysis, design, programming, test, education and installation, can be seen as rituals supporting the view that systems development is a rational process (Robey et. al. 1984).

In line with these conclusions Nandhakumar et. al. report how a system development methodology had the purpose as a symbol to support the fiction of systems development as a control-

lable process, but in use was too structured and rational to be of any help in the development process (Nandhakumar et. al. 1999). Likewise it has been reported that the use of a methodology was found to serve as a social defence against engaging in the real and complicated task of developing the information system (Wastell 1996).

In a study Bansler et. al. found that the developers selected some parts and techniques from the method and combined these with other tools to allow the use of the method to fit their needs thereby circumventing the limitations of the method (Bansler et. al. 1993).

In another study Fitzgerald found that the different methods are used in a pragmatic way resulting in a unique instantiation of the method for each development project. Fitzgerald explains this as the adoption of techniques of the method but without any adoption of the philosophy on which the method is built (Fitzgerald 1998a).

In a recent study Madsen et. al. report that techniques from the method were used but the framework for the development process proposed by this method was not (Madsen et. al. 2002). They explain the departure from the method in this particular case by a conflict between the iterative process proposed by RUP¹ and the customers' demand for a fixed price contract.

Kautz et. al analyse how insufficient focus on eight different factors had an impact on the adoption of the methodology (Kautz et. al. 1994). They conclude that adoption failed and use of the methodology was abandoned due to the antecedents to the introduction making adoption difficult, unclear mission of the introduction, lack of management support, characteristics of the organization culture, doubts about the usability and validity of the methodology, insufficient training and insecurity with the methodology, no systematic monitoring of the introduction process, and a change process with no participation by the methodology users in the decisions taken.

Curtis et al. found that the productivity and quality of the developed systems are effected by limited domain knowledge, fluctuating requirements and communication breakdown (Curtis et al 1988). Curtis et al. suggest that the system development processes do not support activities to support demands for learning about the domain, the fluctuating nature of requirements and the impact from the environmental and organizational context (Curtis et al. 1988).

3. RESEARCH METHOD

Our study relies upon a case study carried out in three development projects in a large Danish software company. The collecting of data is mainly based on twelve semi-structured qualitative interviews with representatives from the projects as well as from the official method support department.

We have chosen to use a research method, which reflects our intention of investigating how the world behaves instead of investigating if the world behaves in a specific way. This emergent strategy leaves us with no specific hypothesis and suggests we adapt an open method.

With the collected data from the interviews we performed an analysis based on the Grounded Theory (Glaser et. al., 1967), (Strauss et. al., 1998). The Grounded Theory framework describes a way to search relevant topics and relations through three sequential steps, as we here will present the way we implement them.

Open Coding: Is the initial step in which the data or text is opened and the different meanings of the participants statements is sought. All text material is read and interesting quotations are categorized and marked with labels explaining the meanings they express. The result of this process is a large set of different codes or concepts presenting the thoughts, ideas, and meanings the text material represents.

Axial Coding: In this process appropriate categories are found which can describe relations between the different codes or concepts. The underlying meanings behind the different concepts

1 Rational Unified Procoess

are identified and are categorized in main and subcategories which in combination constructs patterns or sets of axis explaining the data material.

Selective Coding: Based on the categories we try to combine these into a coherent image of the data. This image is constructed around the main categories and therefore represents an entire framework of the relevant parts of the topics in focus for the investigation.

In spite of our open and emergent strategy, we agree with Walsham in his warning against ignoring any existing theory (Walsham 1995) and therefore we let ourselves and our study inform from relevant literature.

It is important to keep in mind that the chosen research method relies heavily firstly on the data material which is the participants interpretation and secondly on our perception and interpretation of this interpretation, but when examining the phenomena in social contexts "[...] interpretations must include the perspectives and voices of the people whom we study" (Straus et. al. 1998).

4. CASE DESCRIPTION

This study was conducted in a large Danish software company. The company's customers are both private and public organizations. To guide and assist the development of systems, the company has a quality assurance program and a general development process, which has been ISO-9001 certified. An organizational unit, the method support department, is responsible for describing and introducing methodologies as well as consulting the projects in the choice of methodology and to assist in quality assurance.

Two types of projects can be identified. The first type, the maintenance projects are initiated to maintain and extend or improve the functionality of systems already in use. The second type of projects deal with the development of new systems or the reimplementation of older systems using new technology instead.

4.1 The method support department

The company has several formal methodologies described in guidelines, and the project management chooses from these when a new project is constituted. A move towards component based development has been decided by management approximately five years ago and a CASE tool for supporting this has been introduced by the department along with guidelines describing a customized version of the methodology accompanying the tool.

It is the decision of the project management of the individual projects if they want to acquire the services from the department.

4.2 Project A

Project A develops a reimplementation of an administrative system which uses a component based mainframe- and web-technology. The project consists of 15 people, from which four have been interviewed counting one project manager, one product architect and two developers.

Preceding the commencement of this project another project had initiated an analysis of the system to be developed, but the efforts were abandoned due to the complexity. The component architecture of the system was devised on a "work camp", where a selection of the team members came to an agreement on the architecture of the system. To assist the development of the web-front-end, the project has developed a prototype through an iterative process which has resulted in a final prototype that has been approved by a group of users.

Although the mentioned case tool for component based development is used in the project in conjunction with a web-development application, an alternative approach inspired by XP² has been chosen. The approach consists of a number of heuristics that the participants describe as their development methodology. Using these heuristics the project develops the system in incre-

mental steps of about one to two weeks implementing one small part of the system at a time. The project has decided not to use the any consultancy from the method support department

4.3 Project B

Project B enhances an administrative system used by several customers. The system components consist of a large ERP solution combined with a business rule system and mainframe components. During initiation of the project the project manager in cooperation with a few project members performed an initial analysis of the system to be developed and decided on a development plan constituted by 17 sub-projects divided over two years having 3-4 projects run simultaneously with 20 team members altogether. Five interviews were conducted in this project.

To assist in developing the architecture of the system, the project used a consultant from the method support department to help adapting an architecture-paradigm describing five different views on the construction of the architecture of the system. This paradigm was developed by the method support department and had not been used before.

In one of the sub-projects in which three of the interviews were carried out, a method inspired by OOAD³ had been chosen which one of the experienced developers had used before in other projects. The method had been taken in use with some adjustments to fit the particular needs of the project. These needs were a result of the requirements that had already been defined for the sub-project and of the design requirements dictated by the choice to use the ERP system as a component.

4.4 Project C

Project C develops an administrative system based on a client-server architecture. The project, which is close to being finished, employs 20 people, of whom one has been interviewed.

At the time when the project was initiated, it was decided to adapt the component based CASE tool and to use the development methodology for this tool. The guides describe the activities that should be performed during development ranging from how the application domain should be modeled in a business model to how components should be designed in the CASE tool. The design guidelines are closely linked to the tool and describe how components should be created in the tool and give advice as how the functionality of the tool can be used to specify these components.

5. RESULTS

On the basis of our analysis of the empirical data we conclude that the practical use of system developments methodologies are affected by the following five categories.

Universality

Introduction

Experience

Security

Co-determination

In the following we will describe each of these categories in detail and present how we have found that the practical use of methodologies is affected in the projects.

5.1 Universality

Our study shows that the idea of having one universally applicable methodology - the silver bullet - is not shared among the respondents.

The respondents e.g. mentions that an old system, which is developed by the use of a specific methodology and therefore the architecture and documentation etc. are influenced by the pre-

scribed methods and tools, cannot be remodelled, changed or enhanced without paying respect to that methodology.

When developing and designing large systems the developers have experienced problems with the sequential methodologies because the initial analysis tends to be incorrect or inaccurate when the time of the development tends to be longer. The application domain is changing faster than it is possible to design and develop the large and complex systems. One Systems developer argues:

“Yes... Hmm... we gave that up! We found that we couldn't analyse this area thoroughly, because before we were done with that it has changed too much. hmm.. So actually we ditched the waterfall method and said: It won't do it! We'll never finish this if we do it the 'right' way!”

Also the complexity of the large systems are affecting the usability and thereby the usage of the sequential methodology in the development projects. Some projects are so complex that the cognitive abilities of the developers are not adequate to comprehend all the dimensions at any one time. In some situations this might lead the project to a paralytic state where the analysis just goes on and on without ever being finished. As one of the project leaders expresses it:

“Because there are so many details in this... So if you try to embrace the whole area detailed [...] you'll be dead before you finish.”

To avoid these situations some of the projects have adopted another methodology than the official or supplemented it with alternative methods or techniques. In one project the official methodology is not used at all and instead techniques like vertical and experimental prototyping are used to make the process more iterative. Also the use of end users as consultants for the developers are practised. To some extent this helps the project to make a correct design, but also the end users are found to learn new things about the domain as the systems evolve – another indication of the difficulties with the complete up-front analysis tactic.

An additional reason why the one-size-fits-all-methodology is not feasible concerns the belief among the developers that different tools and platforms advocate different methodologies. E.g. it is commonly accepted that developing using the official CASE tool also involves the adoption of the analysis and design guidelines. In one of the projects the chosen platform is a large ERP solution combined with a business rule system and mainframe components which directly affects the chosen methodology, because the project leader and her developers are not convinced that one of the formal methodologies will fit to this development setup.

Also the direct application of the prescribed methods and techniques is influenced by the development tools. In one project where the development is taking place using the CASE tool prescribed guidelines, the developers experience problems because the development tool is not capable of handling certain modeling techniques. This directly affects the development process in that the developers are “forced” very early in the development process to begin the actual physical modelling – instead of starting with a more abstract business model as intended in the methodology.

5.2 Introduction

Also the introduction of the methodology plays an important role with regard to the adoption of the prescribed methods.

In the case company there is a focus on the formal quality systems and thus every project has to comply with the prescribed formal requirements from the ISO-9001 certification. With regard to the choice of development methodology the formal requirements are less distinct and it is very much up to the individual projects to choose between the available methodologies. But having said this a wish from both the management and the method support department exist for the projects to adopt the CASE tool based development methodology. Of the three projects only one was actually following this methodology. One of the reasons for this lack of adoption is,

that the management did not fully back up behind the introduction of it. One project leader argues:

“[...] the management did not treat it as a shift of paradigm, which it really was. It is actually a shift of paradigm to begin using this [...]”

The introduction of the new methodology and tools were not taken seriously enough and the necessary education and training were not offered the developers. The new tools were installed on the working stations and some courses were arranged, but there were no other information about why the new methodology was introduced and which improvements and advantages the use of it was considered to have on the development process.

In one of the projects another methodology was adopted and in the beginning the project experienced some problems with regards to how to use it in practice. The purpose of the different techniques in the methodology was not clear to the developers and therefore their eagerness to use it was lower. Developers have to have a vision for doing the development a new way or they will have no incentives to change their behaviour. In this project the method support department were closely involved and when the problems of understanding the different techniques arose it was easier for the method experts to assist the developers and thus the adoption became easier. Also the method support department benefited from this close cooperation because it became clearer for them which part of the method documentation was unclear and generally which parts of the methodology could be improved.

5.3 Experience

The third theme we have identified among the developers is their experience with regards to both the development process and the application domain.

The more experienced developers express that it is possible for them to use their domain knowledge in many situations instead of the prescribed analysis methods or techniques and that their ability to overview the application domain makes it possible for them to analyse and design the systems without having to go through formal step by step guides.

Another finding relates to the experienced developers ways of using the tools and techniques that they are in fact using. The use of the methodology is not characterized as a general adoption of the underlying philosophy, but more as a tool box from where appropriate techniques and methods can be found. One architect explains:

“[...] we use just as many methods as everybody else... It is just that we are so old and experienced developers, so we just kind of do it without needing a nice template around it [...]”

This pragmatic use of the methodology has some implications for the adoption of new methodologies as well. When the developers are not adopting the underlying philosophy of the methodology their way of working is not changing when a new methodology is introduced – only more tools are put in their tool box.

On the other hand the newly educated or not as experienced developers express a need for explicated methodologies which can help them learn the case company's development practices faster. Some express that they sometimes are frustrated because it is difficult for them to know how to handle certain development situations. Also they have to rely fully on the more experienced developers who therefore possess an important role with regards to the education of new developers.

5.4 Security

The fourth theme we have identified among the developers are concerning security.

The fact that the developers need to feel secure about the progress and usability of their development work is explicated in several ways. One project used a methodology prescribing the development of a kind of very early prototype to help the developers sketch the business model

of the system. These early prototypes are not intended to represent any final parts of the system, and so the end user who are presented with them cannot recognize any user interfaces. This situation was disliked among the developers who felt, that they need to “please” the end users, so they as well could see some actual progress in the project and so they on an early stage were able to recognize small parts of the final system. In another project the project leader mentions that it is important to have a continuous “drive” in the project, and that this “drive” can be intensified by seeing the customers “applaud” from time to time.

Many of the larger systems developed by the case company are “mission critical” systems and this is reflected in some of the participants remarks about the prescribed methodologies sometimes being abridged because of the time pressure experienced in the projects. Typically the developers would like to have more comments from the end users, but the schedule does not allow it, and the developers instead develop the system as they might think the end users want it. The developers experiences this as frustrating as they cannot use the techniques they themselves have agreed upon.

A more symbolic use of the methodologies we have seen in the case company as well. In the project which where using an agile approach it is experienced that the plans and communication “upwards” in the hierarchy needs to be adjusted to what the project leaders thinks the management wants with respect to feeling secure about the projects performance and continuity. Another example is described in terms of some projects stating that they use the CASE tool, but they are only doing so in a very superficial way. These projects hope to achieve some goodwill from the management because they use the “correct” standards.

5.5 Co-determination

Our study suggest that the use of methodologies in the development also are related to the participants need for co-determination.

The developers urge to participate in the planning of their work conditions has lead to a methodological change as one project has adopted a new methodology which not stems from the method support department. The developers explain that they have heard of XP and that this has lead them to want to try something new and something else. An architect states:

“So we needed to try it. You know, it is just as much for our own... for fun!
Otherwise we wouldn't bother!”

In this special occasion also the less formalized and rigid structure of XP plays a role in the developers likening and their reasoning for using that particular methodology which implies that the developers wants more freedom in their work.

We have seen that in a project where there were a high degree of co-determination also the motivation and the eagerness for the project to success were higher. The developers responsibility implied a certain esprit-de-corps among them which made them more willing to work harder in some situations and also made them want to show the rest of the organization that their way of doing things were good and worked well.

6. COMBINING THE THEMES

The combined view of these five themes can improve our understanding of some earlier identified results and to some extent expand our understanding of methodology use in practice. In the following section we combine the results from the five themes with each other and with existing literature from IS field.

Explicitness and dialogue eases adoption

No universally applicable methodology exists

Methodologies are used symbolically

A move towards use of incremental methodologies

6.1 Explicitness and dialogue eases adoption

Our study suggests that there is a closed connectedness between the developers experience and their ability to adopt new methodologies. The more experienced developers tend to use their own experience and choose the techniques and methods that they have tried before instead of choosing and adopting new methodologies.

We have shown that a poor management support makes it even harder to diffuse new methodologies in the organization when it results in a not clear message or when the necessary resources are not available in the introduction process. This finding is similar to the one reported by Kautz et. al. (Kautz et. al. 1994).

Another important issue is the cooperation with the method support department. We show that when the users and the designers of the methodology work closely together with implementing it, the result is better. Both parties benefited from the process because the developers were able to understand the meaning of the techniques provided and the designers were able to get a better understanding of the developers needs and thus could learn new ways of explaining the methodology or redesign smaller parts of it to better comply to the expressed needs. This finding is to some extent reflected in Kautz et al.'s study, indicating that the involvement also bring responsibility along and thereby supporting the adoption. Kautz et. al. further state that the mission – the goals of the change – need to be clear for the developers, otherwise they cannot see the purpose for the change or they might "invent" their own ideas (Kautz et. al. 1994).

Furthermore some methodologies are found to be more intellectual stimulating than others, and therefore preferred by the developers (Johnson 2002). By working closer together the systems methodology department may easier identify for which methodologies this is the case and why it is so. With this knowledge it should be easier to plan the construction and introduction of new methodologies or changes to the existing. Kautz et. al. made a similar finding (Kautz et. al. 1994).

In addition the explicitness of the methodology is an important issue regarding the adoption of and education in new methodologies. In our study we suggest that it is somewhat problematic educating new staff in methodologies when these are not made explicit. In some of the investigated projects the methodology documentation is very sparse and thus new employees have to rely on their more experienced colleagues which are more reluctant to adopt new methodologies, and thus the diffusion is slowed down.

6.2 No universally applicable methodology exists

We have found the idea of having one universally applicable methodology erroneous. This finding is in line with other researcher's suggestions. Brooks describes that the "silver-bullet" does not exist because of the complexity of the systems and their being part of many different interconnected systems (Brooks 1987). Truex et. al. shares some of these ideas and presents the argument that the development process by nature does not allow itself to be controlled by formal structures, and many methodologies directly include certain phases, steps or techniques to find and eliminate errors stemming from the use of the methodology itself (Truex et. al. 2000). Truex et. al. further states that the development process is a locally situated process which cannot be disengaged from the specific context in which it exist (Truex et. al. 2000).

We show that the implication of the varying experience among the developers are having an impact on the use of methodologies leading to the application of different methodologies in the different projects. This finding is mirrored in other research. Stolterman shows in his study that developers used methodologies only to some extent. They were able to do so because they "just knew" what to use (Stolterman 1992). These findings are also reported by others (Fitzgerald, 1997), (Fitzgerald 1998a), (Bansler et. al. 1993). Fitzgerald reports about a variance by systems developers in their use of methodologies depending on their developer experience. When the developers are "fresh" they tend to follow the methods more rigorously, but after gaining more experience they tend to manage more by themselves and not use the methodologies to the same extent. After gaining even more experience and more insight into the problem areas, and

therefore more easily can see why the methodologies can help them, the developers again use the offers from the methodologies (Fitzgerald 1997), (Fitzgerald 1998a).

Bansler et. al. find that some developers did not use a methodology as specified, because they believed that some of the techniques would not give them any new information because they knew the system well (Bansler et. al. 1993).

Our findings suggest that the developers are inspired to apply new methodologies they have met outside the organization e.g. in professional networks or the like and thus from time to time introduce new methodologies or parts hereof into their project work. This will sometimes eventually lead the projects to adopt new practices, and thus the need to try something new or different makes the idea of a universally applicable methodology impossible. In line with this finding Fitzgerald finds in his study, that working with new technologies was cited as a significant motivator among developers (Fitzgerald 1998b). In accordance with this, Stolterman notices how some developers become irritated when a certain method do not credit the developer with a will and ability of his own to judge in the specific situation and therefore a methodology has to allow a certain element of freedom of choice (Stolterman 1992).

When the developers express that they see a great interconnectedness between the choice of development platform and the choice of methodology this also indicate that more than one methodology is needed if more than one technical setup is used. Agreeing with this Orlikowsky et. al. report from a study where the development tools constrained the development work and limited the extent and nature of the developer autonomy thus restricting the design vocabulary to concepts "known" to the tools (Orlikowsky et. al. 1991). Also Nandhakumar et. al. report that a software tool can force the developers to use specific techniques, and thus letting the choice of tool be the same as choice of methodology (Nandhakumar et. al. 1999).

6.3 Methodologies are used symbolically

Our study provides examples of the use of methodologies with the purpose of making different parties more secure regard to the project progress. This kind of use of methodologies is also reported by Wastell, who in a study found that developers sometimes use the methods prescribed by methodologies just because, they are supposed to do so. The system they are developing does not really benefit from the methods, but the developer has done "the right thing" and therefore are "secure" (Wastell 1996). A finding also reported by Nandhakumar et. al. (Nandhakumar et. al. 1999)

Another twist of this argument is that the methodology can help the developers overcome the rather difficult task of developing large information systems. The methodology can act as a means to provide the developers the necessary comfort to become safe and to make them believe that they are capable of the task (Wastell 1996).

Also Robey et. al. gives examples of the symbolic or political use of methodologies in that they find certain processes performed only to give the overall development process a rational appearance (Robey et. al. 1984).

6.4 A move towards use of incremental methodologies

Our study suggests that there is a move towards using methodologies prescribing an incremental work flow and away from the sequentially organized methodologies. This is in line with many other studies which give various reasons for this trend.

Like Fitzgerald we find that the rapid changes in the application domain and the changing nature of the business environment makes it inappropriate to base development on the traditional life cycle approaches (Fitzgerald 1998b).

To this argument we add that an iterative approach might give the developers as well as the clients a sense of control and comfort meaning that the continuous deliverances or tests of prototypes directly illustrates to both parties the progress of the project. This point is further developed when also considering the extra motivation we have observed among the participants

when they are able to see that their work can be helpful to the clients. On the other hand this shift towards the adoption of incremental methodologies is not seen as an entirely positive shift from the customers point of view. Many customers want control over the process and therefore fixed prices and fixed delivery times can be a problem when working with incremental methodologies (Madsen et. al. 2002), (Bjerknes et. al. 2000).

Also the size of the systems can become so large that developers are not able to cognitively comprehend them, and thus it is necessary to break them up in smaller parts or to start working with a smaller subset – e.g. like prototyping. The same argument is seen to be valid for the customers or end users – they cannot comprehend the total system at once either (Brooks 1987) (Curtis et. al. 1988).

7. CONCLUSION

The results of our study confirm existing research in the field and offer some new insights, especially with regard to the interplay and relationship of the different, identified themes.

We confirm the relationship between system developers' experience in systems development and the pragmatic utilization of methods and add that experienced developers value their experience as more significant than the possible adoption of new methods, which also explains the lack of method adoption. In addition, we substantiate the importance with which strong management support has for the adoption of methods. We also show that a close cooperation between method developers and method users as well as experienced developers' influence on their less experienced colleagues have an impact on method adoption.

We identify the necessity of developers' access to a selection of methods, which confirms the methods' flawed universality. We provide an explanation and a deeper understanding for the need for various methods and differing method components in a development organization and relate this to the developers' wish and need for involvement in the method selection and development as well as to the close relationship between the technical platform, development tools and methods.

Our study confirms the symbolic function a method may have and relates this to the perceived need of the management for security with regard to the development processes, which is satisfied by the development of a documented method.

The study provides further evidence for the misconception of universally applicable methods and extends the understanding of the systems development shift towards incremental methods and towards situated use of methods. We confirm that the development context's volatility and a project's apparent uniqueness are reasons for this shift.

We also show that a system's size and complexity can be in conflict with cognitive limitations with regard to the use of comprehensive methods. Furthermore, the shift is related to the perceived need for the creation of security of and between the parties involved concerning the course of a development project. Finally, formation and maintenance of developers' motivation is associated with this shift.

Thus, our work provides an empirical study, which confirms and extends existing knowledge in the field and offers an empirical documentation, which can be exploited for further research on the subject. The most important contributing is the empirically verification of some of the findings which enlarges the body of knowledge.

REFERENCES

- Avison, D.E. & Fitzgerald, G. *Information Systems Development: Methodologies, Techniques and Tools*, McGraw-Hill, Maidenhead, UK, 1995
- Bansler, J. & Bødker K. "A Reappraisal of Structured Analysis: Design in an Organizational Context", *ACM Transactions on Information Systems*, (11: 2), 1993, pp. 165-193
- Bjerknes, G. & Mathiassen, L. "Improving the Customer-Supplier Relation in IT Development", *Proceedings of the 33rd Hawaii International Conference on Systems Sciences*, 2000
- Brooks, F. B. jr. "No Silver Bullet - Essence and Accidents of Software Engineering", *Computer*, (20:4), 1987, pp. 10-19
- Curtis, B., Krasner, H. & Iscoe, N. "A Field Study of the Software Design Process for Large Systems", *Communications of the ACM*, (31:11), 1998
- Fitzgerald, B. "The Use of Systems Development Methodologies in Practice: A Field Study", *The Information Systems Journal*, (7:3), 1997, pp. 201-212
- Fitzgerald, B. "An Empirical Investigation into the Adoption of Systems Development Methodologies", *Information and Management*, (34), 1998a, pp. 317-328
- Fitzgerald, B. "An Empirically-Grounded Framework for the Information Systems Development Process", *Proceedings of ICIS (International Conference on Information Systems)*, 1998b, Helsinki, Finland
- Glaser, B.G., Strauss, A.L. *The Discovery of Grounded Theory - Strategies for Qualitative Research*, Aldine De Gruyter, New York, 1967
- Johnson, R.A. "Object-Oriented Systems Development: A Review of Empirical Research", *Communications of the Association for Information Systems*, (8), 2002, pp. 65-81
- Kautz, K., Malmborg, L., Pries-Heje, J. "Does University Education Lead to Adoption?" in T. J. Larsen et al. (eds.) *Information Systems: Current Issues and Future Changes*, IFIP publications, Laxenburg, Austria, 1998, pp. 559-576
- Madsen S. & Kautz K. "Applying System Development Methods in Practice - The RUP example", *Proceedings of 11th International Conference on Information Systems Developments, Methods & Tools - Theory & Practice*, Riga, Latvia, 2002
- Nandhakumar, J. & Avison, D.E. "The fiction of methodological development: a field study of information systems development", *Information Technology & People*, (12:2), 1999, pp. 176-191
- Orlikowsky, W.J. & Robey, D. "Information Technology and the Structuring of Organizations", *Information Systems Research*, (2:2), 1991, pp. 143-169
- Robey, D. & Markus M.L. "Rituals in Information System Design", *MIS Quarterly*, (8:1), 1984, pp. 5-15
- Stolterman, E. "How system designers think about design and methods. Some Reflections Based on an Interview Study", *Scandinavian Journal of Information Systems*, (4), 1992
- Strauss, A. & Corbin, J. *Basics of Qualitative Research*, SAGE Publications, London, UK, 1998
- Truex, D., Baskerville, R. & Travis, J. "Amethodical systems development: the deferred meaning of systems development methods", *Acting., Mgmt. & Info. Tech.*, (10), 2000, pp. 53-79
- Walsham, G. "Interpretive case studies in IS research: nature and method", *European journal of Information Systems*, (4:2), 1995, pp. 74-81

Wastell D.G. "The fetish of technique: methodology as a social defense", *Information Systems Journal*, (6), 1996, pp. 25-49