The Utilization of IS Development Methodologies in Practice

Karlheinz Kautz, Bo Hansen, Dan Jacobsen
Copenhagen Business School, Department of Informatics, Njalsgade 80, DK-2300 Copenhagen, Denmark

ABSTRACT

The objective of our research is to study people and their actions within an organizational context, namely whether and how system developers use Information Systems Development methodologies in practice. This article adds to the ongoing discussion about the practical use of ISD methodologies. Many prescriptive and normative textbooks as well as anecdotes exist, but there is little empirical documentation about the actual use of development methods. To contribute to the scientific documentation of methodology utilization, we provide here an empirically based study of the practical use of development methods in three projects within a large Danish software development company. Founded on a Grounded Theory analysis we identified five main categories of themes, which are related and have an effect on the use of ISD methodologies. These include 1) the extent to which a specific methodology is universally applicable, 2) the need for confidence of and between the parties involved, 3) the degree of the developer's experience, 4) the developer's wish for co-determination in the development process, and 5) the nature of the introduction process for a method. Our results confirm existing research in the field and offer new insights, especially with regard to the interplay and relationship of the different, identified themes.

Keywords: ISD methodologies, Grounded Theory, case study, method universality, confidence, experience, co-determination, introduction

INTRODUCTION

This article contributes to the discussion about the practical use of ISD methodologies. The literature on IS development generally, and IS development methodologies in particular is extensive and wide ranging. It consists however largely of prescriptive and normative textbooks and work that is based on anecdotes, but there is limited scientifically collected and analyzed empirical documentation about the actual use of development methods (Nandhakumar and Avison, 1999).

A number of arguments for adopting and using ISD methodologies have been put forward. Avison and Fitzgerald (1995) mention the accurate recording of requirements, the possibility to monitor progress to identify changes as early as possible, the delivery of systems within appropriate time and cost limits, securing well documented systems easy to maintain, and making it possible to deliver systems which are appreciated by the relevant parties. Fitzgerald (1998a) extends this list to 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 the possible standardization of the process and thereby the facilitation of the interchangeability of developers among projects.

In spite of these arguments various studies report that the methodologies are often not used as intended and that systems developers in projects either question the purpose of the methods and techniques specified in
the methodologies or express a need for a different kind of support for their 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 and Bødker, 1993), (Fitzgerald, 1998a), (Madsen and Kautz, 2002). Little research is however done about what influences the actual utilization of methodologies. Exceptions are Fitzgerald et al. (2002) who put forward a method-in-action framework, Vidgen et al. (2004) who provide a historical account of the dynamics of methodology emergence in a development project that lasted 2 years, and Kautz (2004) who presents a framework and describes the enactment of methodology in a Multimedia information systems development project.

To further fill this gap and contribute to the scientific documentation of methodology utilization, we provide an empirically grounded study of the practical use of development methods in three projects within a large Danish software development company. As such, the objective of our research is to study people and their actions within an organizational context, namely whether and how system developers use Information Systems Development methodologies in practice. The article is structured as follows: Section 2 presents the literature findings that helped us understand and conceptualize this study and section 3 describes our research method. In section 4 the case company is introduced and in section 5 our main results from the analysis are presented. Section 6 contains a discussion of these results and emphasizes our main findings, and finally section 7 summarizes our study.

BACKGROUND

Research on applying ISD methodologies suggests there is a disparity between the way methods are formally described and the way in which methods are used in practice. Truex et al. (2000) argue that the assumptions underlying the concept of ISD methodologies must be addressed. Based on extensive filed work and literature studies they question that ISD is a manageable, linear, repeatable and rational process and propose an alternative set of amethodical assumptions, but they also suggest that both views must be kept in mind when engaging in the development of information systems. Stolterman’s work (1992) supports that the assumption underlying the perceived need for ISD methodologies is that a developer’s basic approach to developing systems is irrational and that this irrationality must be addressed to let the process follow the ideal of rationality. Robey and Markus (1984) propose a political view on ISD methodologies and describe how the activities in traditional system development life cycles can be seen as rituals supporting the view that systems development is a rational process. In line with these conclusions Nandhakumar and Avison (1999) report how an ISD methodology had the purpose as a symbol to support the fiction of systems development as a controllable process, but in use was too structured and rational to be of any help in the development process. Likewise, Wastell (1996) reported that the use of a methodology served as a social defense against engaging in the real and complicated task of developing information systems.

Bansler and Bødker (1993) found that developers selected some parts and techniques from a 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. Fitzgerald (1998a) also found that different methods are used in a pragmatic way resulting in a unique instantiation of the method for each development project. He explains this as the adoption of techniques of the method but without any adoption of the philosophy on which the method is built. Madsen and Kautz (2002) report that techniques from a method were used but the framework for the development process proposed by this method was not. They explain the departure from the method in their particular case by a conflict between the iterative process proposed by the used methodology and the customers’ demand for a fixed price contract.

JITCA, Volume 6, Number 4  2
The Utilization of IS Development Methodologies in Practice

Kautz and McMaster (1994) analyzed the introduction of a structured methodology and identified eight different factors that had an impact on the adoption of the methodology. The adoption failed and use of the methodology was abandoned due to the antecedents to the introduction making adoption difficult, an unclear mission of the introduction, a lack of management support, hostile 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. Finally, Curtis et al. (1988) found that the productivity of the developers and quality of developed systems were affected by limited domain knowledge, fluctuating requirements and communication breakdowns. They argue that system development processes and methods do not support learning about a domain, understanding the fluctuating nature of requirements and realizing the impact of the environmental and organizational context, in which systems development takes place.

RESEARCH METHOD

Our research relies upon a case study carried out in a large Danish software company. To secure the validity of our work we largely followed Klein and Myers's (1999) seven principles for interpretive field research and Eisenhardt's (1989) seven steps for case study research. The data collection is based on twelve semi-structured qualitative interviews.

Considering the resources available to our study, we had to concentrate on a single case. Our results are thus based on the method application in one selected system development organization. This approach might be criticized for only generating a local empirical theory, which might not be generalisable, but as argued by Hughes and Jones (2003) it still is useful and contributes to the existing body of knowledge. On the other hand by limiting the study to a single organization, we were able to examine the case in more detail to understand more thoroughly the interrelationships of separate data. Due to the project's limited resources, we would not have been able to achieve the same depth in our examination if we had chosen to study several case organizations.

As the subject of the analysis deals with systems development methods in practice, the analysis took place in an organization experienced in the application of such methods. The organization had used methods long enough to have incorporated method application in the daily routines and to have gained experience in method application. To achieve a broad-spectrum study, we carried our interviews for three different projects from different departments in the organization. The interview respondents represented the roles in the system development organization previous studies had indicated as important for method application: systems developers, project managers and staff in charge of method development and training. We interviewed both newly qualified and experienced systems developers. We chose a research method, which reflected our intention of investigating how the world behaves instead of investigating if the world behaves in a specific way. This emergent strategy left us with no specific hypothesis and suggested that we adopted an inductive and open method.

Qualitative Interviews

As we have chosen an open approach to the subject, the framework in connection with the data collection could not be fixed. Therefore, we focused on performing semi-structured, qualitative interviews with the personnel from the case organization. The situation in the development department is characterized by the employees' interpretation of their normal everyday life – there is no objectivity, only subjective manifestations of interpretations and our subjective interpretation of these manifestations. Therefore, our
data collection is qualitative in nature, as we cannot describe human relations and the like solely by means of quantifiable parameters.

The interview guides were developed concurrently taking advantage of interesting data that we came across during the interviews and the document analysis. There were a total of twelve interviews; all performed during one month. The majority of interviews lasted an hour and a half, a few lasted an hour and one interview took two hours. Two researchers participated in all the interviews, which were conducted with one respondent at a time. The interview roles were decided in advance. One researcher functioned as the main interviewer while the other supplemented when the interview came upon topics that were worth pursuing. All interviews were recorded on tape, as the application of Grounded Theory requires detailed data collection. It is precisely through the abundance of data that relations and conclusions emerge. Therefore, it is vital for a case study like ours to capture the respondents' interpretations very precisely and ensure that the social interaction in the interview does not suffer (Walsham, 1995).

We chose tape-recording rather than taking notes of the respondents' answers. Taking notes would invariably involve an interpretation and a selection of what to write down and what appears to be relevant in the interview situation. By taking notes, we might miss statements that later may turn out to be relevant. However, tape-recording the interviews also has drawbacks as the respondent might be less inclined to speak openly about sensitive topics when the interview is recorded (Walsham, 1995). To reduce this effect, we used the first minutes of the interview to make the atmosphere more congenial. As a result the respondents expressed that they did not feel unsettled, as the topics of the study never dealt with sensitive personal information. All the recorded interviews were transcribed providing us with the full interview text. A complete transcription of the taped interviews is very time consuming, but we chose this solution to avoid the loss of data mentioned above. The word-for-word transcriptions constituted approximately 340 standard pages of text and took approximately one month of work. In this way, we were able to use the interview transcripts directly for the coding of data in the subsequent analysis phase. Moreover, by studying the full interview texts, we got a thorough repetition of all elements in the interviews, which improved our insight into the data material.

The questions in the individual interviews were based on existing theory within the subject, and for this purpose we worked out interview guides for the individual interviews. As we worked with semi-structured interviews, the interview guide was used as a checklist for issues that had to be covered during the interview and not as an actual outline for the interviews. The nature of the interviews had been open, and when the conversation moved towards new and interesting areas relevant to the subject, we pursued and probed the new directions.

To supplement the interviews, we studied documents made available by the case organization. These studies also inspired us in preparing the interview guides. The study of official documents and directions from the case organization helped us acquire first hand knowledge of the organization, its formal regulations and especially of the method application and documentation. This gave us the opportunity to learn about these issues before starting the interviews and allowed us to examine whether descriptions of formal system development methods existed and to determine which type of documentation was available to the employees. The examination of documents, i.e. standards, directions, recommendations and method descriptions, supplemented our background knowledge prior to the interview rounds. The concepts and descriptions in the documents made it easier for us to understand the respondents in the interviews and to relate to the use of the methods described in the documents.
The Utilization of IS Development Methodologies in Practice

We actively used the information from the document studies to prepare the interview guides. Already in the first interview, this enabled us to inquire about interesting phenomena in system development in relation to the documented information. The document studies enabled us to assess the materials available to employees working with system development and how much support they could get from these materials. The studies made it easier to evaluate to which extent such advice and directions were followed and which effect the recommendations had on the final implementation of methods in the projects. At the start of the project, we selected both electronic and paper documents in cooperation with a contact person the organization had assigned to our project. However, as we gradually learned about other documents that might be interesting through conversations with our informants, this material was also used in our further data collection efforts.

Applying Grounded Theory

With the collected data from the interviews we thus performed an analysis based on the Grounded Theory methodology (Glaser and Strauss, 1967), (Strauss and Corbin, 1998). The Grounded Theory framework describes a way to search relevant topics and relations through three sequential steps:

The purpose of open coding is to open the data material. This is done by looking for different meanings in the statements and classifying part-statements with labels to explain the meanings of the different parts. The result is a range of different codes and concepts comprising the thoughts, ideas and meanings of the text. The purpose of axial coding is to find the categories into which the discovered codes/concepts can be classified. The meanings behind the concepts are compared and categorized in main and subcategories or concepts, which together present patterns or a set of axes to explain the data material and relationships between the concepts. Thus, the located axes reflect the parameters that are important for the study’s subject. Based on the axes categories, the purpose of selective coding is to explain relationships and contexts to refine the overall explanation into a coherent picture of our observations. The overall picture is based on central categories and represents a complete framework of explanations for the field in focus.

As our analysis following the data collection drew on Grounded Theory, we found it relevant to use techniques that could help us to interpret our data during the data collection. Therefore, the transition from the data collection phase to the analysis phase was not clearly defined, which is why the presentation below also includes information originating in the data collection phase.

We used an approach to the analysis that in practice allowed us to apply the analysis techniques open coding, axial coding and selective coding from Grounded Theory. In general, we can describe our approach as follows: In parallel with the actual data collection during the interviews, we reflected on possible relations, which the two researchers subsequently discussed and compared. In a joint brainstorm session, we wrote down our immediate impressions and ideas. With the results from the brainstorm in mind, we individually read and together coded all the interview transcripts and subsequently gathered the codes. Then, we searched for coincidences and patterns in the codes and evaluated them in relation to each other. As a result, we achieved a general picture of the statements from the interviews grouped around important axes and categories in the data material. Subsequently, we arranged the categories into main and subcategories and described and explained the main categories based on the data material.

The following description covers the result of a long and comprehensive process. As the design of the study drew on previous theories within the area, we had a broad theoretical foundation for our interviews from the very beginning. Therefore, we were able to make direct inquiries into areas, which previous studies had indicated might be relevant to system development in practice. The purpose has not been to validate previous theories, but rather to increase our understanding of the area. Thus, our analysis started
simultaneously with the interviews, which is why the data collection process can be characterized as iterative in nature and incorporates ongoing reflections on previously collected data, a process termed constant comparison by Strauss and Corbin (1998). For example, in the individual interviews we pursued issues that suddenly came up and which we assessed to be relevant for further clarification. As mentioned above, we continuously adjusted the interview guides throughout the data collection phase. If we came upon new information in an interview, we added new questions to the interview guides, which other respondents subsequently could elaborate on. In other words, we included new questions when we received information, on which we would like to hear other respondents’ views, or if we conceived an idea that we believed might be relevant to the overall study.

Such ideas were also developed in the process of transcribing the individual interviews, as we did the transcription concurrently between the interview rounds. As a result, we already had a broad and comprehensive insight into the respondents’ statements and an idea of how to understand the interrelationships in the organization prior to the start of the actual analysis. Simultaneously with gaining an impression of the development process in the projects during the interview phase and the transcription of the interviews, we generated ideas about matters pertaining to method application. Prior to the in-depth analysis based on these ideas, we were able to make a joint brainstorm to find and identify candidates for categories of relations that might apply to the system development processes. In the brainstorm, we listed all the different relationships and ideas about relations that immediately came to mind, discussed them and organized them. Through this exercise we reached a first consensus, which we used in the subsequent in-depth analysis.

The brainstorm produced a number of claims and statements constituting the immediate relations perceived during our conversations with the respondents. At the time, we could not call these statements for final or correct, as we had not related them directly to data. But we used them in the analysis as starting point for our further work. We made an open coding of the interviews in order to deduce candidates for categories. Therefore, we took our point of departure from the interview transcripts, which we individually and together read a couple of times. As mentioned above, we chose to transcribe the tape recordings completely, which meant that we were able to see both our own questions and the respondents’ answers. This enabled us to compare the individual statements with the context in which they took place and consequently to see which question or part of the conversation had led to which answer. While reading the interview transcripts, we made codes/notes in the margin referring to the statements that we assessed as important for the respondent’s understanding of system development and method application.

After coding the interview transcripts, we started on the subsequent phase aiming to get a perspective on the coding made in the individual interviews. To this purpose, we used a poster measuring 1½ x 1½ meters on which we transferred all the coding from each interview grouped under each respondent. We gave the respondents a color code based on the project to which they belonged. This measure made it easier for us to distinguish between statements from different respondents later on in the process. We also gave each coding a number and marked these with a reference to the page number of the transcript from which the code and thus the source came. This measure enabled us to find statements that we wanted to check or see in which context they took place.

It was also important to be able to trace conclusions back to the three projects to concurrently see whether the results derived from a single project’s data or might apply to several projects. Therefore, we grouped the codes referring to the individual respondents around the projects to which the respondents belonged. This allowed us to relate the individual results to matters pertaining to a specific project or to matters pertaining to system development processes in general. This visual recording of codes allowed us to identify the
source of the statements and simultaneously to view all the statements in one place, which subsequently made it easier to find relationships between the different respondents' statements. Based on the codes, we proceeded in an exceedingly iterative process and created a range of categories, under which we could fit the different meanings behind the codes.

In practice, we made a new poster with the same dimensions as the first one, to which we transferred the number code representing each individual statement from the respondents. We used the new poster to categorize the individual coding into categories and subcategories. In the first poster, we grouped the statements around the respondents to whom they belonged. The second poster was used to find the general relationships between the many statements, and therefore we placed the codes from the supporting statements around the general relationships.

Each code from the first poster with the respondents' statements was placed on the second poster indicating the respondent to which the code belonged and the number of the code. The indication was made in the color assigned to the project in question. This form of notation ensured that we subsequently could find the relationship with the respondent's statement in the source material for verification or to amplify the information about the context, in which the statement was made. To group similar statements, which appeared in the interviews, we placed the attached code number, respondent, and color alongside already transferred codes. This approach to color-coding enabled us to see which categories were based on data from several projects and which were derived from a single project. In this way, we were able to check whether the category applied to the context of a single project or to all three projects in general.

In connection with the transfer of codes, we placed codes that were related to each other or belonged to primary or subordinate categories in such a way that the relationship was indicated by how they were placed in relation to each other. In term of Grounded Theory (Strauss and Corbin, 1998) we applied paradigm modeling, an analytical tool to integrate the conditional context structure, in which a category is situated, with the sequences of (inter)action processes that pertain to a phenomenon. The work involved complicated analyses and interpretations, including frequent conversations, discussions and thorough examinations of the many statements and their contexts, and it led to many iterations and changes. As a result, we achieved a visual picture of the many codes grouped according to the content of the statements. This enabled us to assess which statements we believed contributed to clarifying the importance of the different categories. The results from the poster were written down and grouped according to 1) phenomena, 2) the causes and the contexts of the phenomena, 3) the actions performed by the persons under these conditions and 4) the intended and unintended consequences of these actions. Thus, we produced a list of the results derived from the texts.

Therefore, the last part of the analysis was based on the list of results describing partly the phenomena encountered in the interviews and partly the influence of the phenomena on the development situation in the projects. The objective of selective coding in Grounded Theory is to select a key category or a theme that can be used as a central focus for the results of a study (Strauss and Corbin, 1998). We chose to use several themes as our data indicated more than one significant theme. To this purpose, we used the list of results, cutting out the individual lines and placing them on a new poster, where they were grouped according to which main themes they appeared to fit. Once more, the process was iterative, as the main themes emerged as a direct consequence of our efforts to place the results. The different constellations of results affected the creation of ideas and led us to identify new main categories. The process stopped when all the results had been placed under the themes that we believed gave the best description of the relations between the results. Subsequently, the main themes were used to structure the description of the study's results. The resulting poster was used directly to formulate the outcome of the study. We listed the results based on the main
themes and described the themes by means of related results. As each coding referred to a coding in the interview transcripts, we were able to recover the original interview quotations that led to the coding and used them in the description of the achieved results. The analysis produced 5 categories with 16 sub-categories, which can explain whether, how and why information systems development methodologies are used in practice. They are described in a 150 pages report and were first discussed in Hansen et al. (2003).

In spite of our open and emergent strategy and Glaser’s (1992) advice that there is no need to review any literature in the area under study so as not to ‘contaminate’ the researchers’ interpretation of data, we followed Walsham’s (1995) warning against ignoring existing theory completely and Strauss and Corbin’s (1998) suggestion to build up a certain theoretical sensitivity. Thus, like others (Fitzgerald, 1998), (Hughes and Jones, 2003) we let earlier literature inform us about the subject, which is in line with interpretive case studies (Walsham, 1995) and we have been inspired by existing literature within the area benefiting from previous results achieved by others. The primary reason for going back to existing literature was the extent of our study, in which it might be difficult to gather enough data to examine all sides and aspects of the subject. We wished to achieve an interaction between the existing literature and our observations from the case organization to explain interrelationships and contribute to the theories with new insight from practice. We in particular used the literature studies to prepare the interview guides, on which the interviews in the organization were based.

This raises the question whether we have used Grounded Theory correctly (Hughes and Jones, 2003), (Strauss and Corbin, 1998) or whether we exposed ourselves to the risk of ‘contaminating’ our analysis. However as Walsham (1995) comments, “It is possible to access existing knowledge of theory in a particular subject domain without being trapped in the view that it represents the final truth in that area”. Thus we were aware of the threat and kept our minds open. The existing literature informed our work and supported our theoretical sensitivity, but we did not get trapped by it and the knowledge it represented.

It is important to keep in mind that the chosen research method relies heavily on the data material which is the participants’ interpretation of the phenomenon under investigation and on the researchers’ perception and interpretation of this interpretation, but in line with Strauss and Corbin (1998) we acknowledge and cannot avoid such a situation when examining phenomena in social contexts where "[...] interpretations must include the perspectives and voices of the people whom we study". Our research approach is explained in more detail in Hansen and Kautz (2005).

CASE DESCRIPTION

The study was conducted in a large Danish consultancy and software company with 2300 employees of which 620 are system developers. The company was founded in 1972 by National Association of Local Authorities in Denmark. It delivers software solutions mainly for the public sector but also for private customers. The solutions consist of different systems which can be used to manage and administer the current legislation as well as for the daily administration and resource management within different areas. As an overall coordinator and resource allocation unit the company has a business team, which chooses and prioritizes the company’s projects.

Systems development in the company is primarily directed towards product development and adjustment and less towards the development of customer specific solutions. As the customers to a large extent have uniform needs, because their work tasks follow the legislation, it is possible to develop a form of off-the-shelf standard products, which subsequently can be adjusted to concrete the use situations of the different customers. The development and maintenance of the individual systems is taken care of by projects, which are responsible for whole of parts of the systems. As a consequence of the different systems’ size and nature, the size and composition of the projects varies to a large extent. Therefore a typical project as such does not exist.
There are two principal groups of projects, new development and maintenance projects. The task of new development projects is to build new system, in reality they are however renewal and reimplementation projects, as it is not too often systems for total new application areas are to be developed. Instead the existing systems are modernized in intervals of 8 - 12 years to secure that they still live up to the customers' requirements with regard to the precision needed to solve the user tasks and with respect to the demanded user friendliness. It is a characteristic of the company's product that they are seldom changed radically. The systems are built for the use in public administration and to deal with administrative tasks, and these areas are rarely exposed to extensive changes, but are instead continuously influenced by changes of legislation in the matters they are supposed to support.

New development projects often deal with existing systems' shift of technical platform, e. g. for many systems there is a currently a strong focus on porting parts of their functionality to web-based platforms. New development projects are usually divided into smaller parts, such that the systems are gradually, part for part, renewed. The most important task of system renewal is to secure that the system are continuously legal, which means that they at any time are adjusted to the laws, which are administered through them.

The company uses a number of technical platforms, methods and tools for the development and operation of their products, e. g. database products from several leading provides as well as different programming languages and 4. generation CASE tools. A standard for the choice of application tools does however not exist. As however more and more systems are developed, which are to be used through a web-based interface, tools supporting the development of such interfaces have been introduced. The company's management has expressed the wish that they increasingly would like projects to use existing components for application development. Therefore approximately 4 years ago 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. Beyond the method documentation of the CASE tool related method, the company has several methodologies formally described in guidelines and developed by the method support department, which comprise both structured and object oriented analysis and design techniques based on a waterfall model inspired overall development scheme, and ideally project management chooses from these when a new project is constituted.

The company is strongly involved in work with quality management in the projects and many of the requirements, which the projects have to fulfill with regards to rules and procedures, are directly derived from the company's ISO-9001 certified quality system for product development and product phase out. The quality system among others requests for example the production of formal documents with regard to project start, requirement specifications, milestone and documents reviews. As the projects can not change these demands, they have to be considered as given for any project.

The Method Support Department

To support and guide the projects in their development work, the company has a special method support department. The department functions as consultant for the projects, which can 'buy' help for all kinds of methodological problems they encounter. It is the decision of the project management of the individual projects whether they want to acquire the services from the department. The method support department is part of a larger support unit which beyond project management and method support among others also offers consultancy in usability questions and quality management. On demand the department arranges also training courses for the projects. This department is responsible for working out method guides and guidelines and to develop methods, which can be used for project management and systems development. The department has to secure the continuous development of the company's methods such that these live up to the demands which are posed by different interest groups, e.g. management or development projects. All
projects receive the offer to get a system architect, who is an employee of the method support department, connected to their project. A system architect can help to secure that the system complies with the technical standards of the company and a system architect can offer guidance with regard to the structure of the product architecture. This is important as many systems have interfaces with already existing systems, not least back-end systems. The system architects also collect feedback with regard to architectural solutions.

For those projects, which apply the CASE tool, the department has developed analysis and design guidelines, which describe a systems development methodology and how it can be used in connection with the use of the tool. These guidelines together with other advice and instructions are all accessible for the projects through the company’s intranet. Based on a specifically developed interview guide investigating their role in the company and in the projects with regard to method utilization we interviewed 2 method consultants. One of them has been in the organization for many years and has been the designer and developer of a number of the applied methods and guidelines, whereas the other has been hired recently and is an expert in a specific type of development tool.

The Investigated Projects

Project A develops a reimplementation of an administrative system, which is used by large parts of the public sector in Denmark. At the time of investigation the project had been running for a year and just delivered its first major release. The project consists of 15 people, of which 4 have been interviewed including the project manager, the product architect and 2 developers.

The system which has to be redeveloped is over 20 years old and has a large amount of users. Therefore, the system has to be in operation during the whole development process. While the old system will be renewed totally and phased out gradually, another project maintains and operates those parts, which are necessary for daily operation. As the project is a redevelopment project and no drastic changes are planned, the basic functionality is known. The project’s main task is to shift out the product architecture and the interfaces of the system. However, the project has to take into account a serious change of legislation, which will be effective in one year’s time. This change of law has to be handled solemnly by the new system, as no old back-up systems exist. Implementing this change on time is of uttermost importance and the project’s first real test.

The system is divided into three layers involving back-end systems, a process layer and dialogue components, which in the future should function on a web-based platform. To perform the development work the project mainly uses the CASE tool, a web development tool, and a – for the developers – well-known mainframe programming tool. Project management has however decided to use a development methodology, which has not been used in the organization before. To achieve company management’s acceptance they have developed a method description, which explains the method. On the background of this document, management has given their accept. There has been an earlier attempt to re-implement the system, which however never got beyond the analysis phase. This failed attempt is considered to be one of the reasons why management approved the project’s alternative approach.

The project started with devising the component architecture of the system on a “work camp”, where a selection of the team members came to an agreement on the architecture of the system. Subsequently the project has specified the system requirements on an overall level with the help of use cases. Based on these documents and a general prototype, project management has identified a number of sub-projects. These last typically between 1 and 2 weeks and solve a small, but well-defined part of the whole problem.
To develop these sub-systems the project uses an approach, which is inspired by the extreme programming approach XP (Beck, 1999). The above-mentioned method document describes a number of heuristics and principles for planning and performing the development work, which the project participants themselves refer to as the method the project follows. The applied principles are: solve the most complex problem first; do not write complete requirement specifications, involve the users; overall contracts and agreements are sufficient, trust is better than control; always strive for simplest solution; continuous testing allows for short development circles.

The project has chosen not to cooperate with the method support department, as both project management and the developers themselves take on the task of developing and organizing the utilization of the method. The project has a strong emphasized on the fact that it does not use one of the usual method and it therefore pays much attention to following the approved heuristics. The interviewed project participants themselves believe that their alternative approach is one important reason why the project reached its first larger release on time.

Project B is a replacement projects that renews and enhances a large administrative system, which is used for staff and personnel management by many public and private sector organizations. Company management has defined the project as a high priority endeavor with significant strategic and business value for the organization. Thus it enjoys a prominent status which not least is expressed through generous resource allocation.

The actual replacement will take place in form of a pilot operation, where the new system and the old one will for a period run in parallel until the implementation of all requirements and stable operations is secured. The project is one of the largest ones in the organization. Thus during its initiation project management together with a few product architects performed an overall analysis which resulted in a development plan for the whole replacement project through 17 sub-projects to be performed over a period of 2 years having 3-4 projects run simultaneously with 20 team members altogether. We interviewed the overall project manager, 1 product architect who works across the sub-projects, on sub-project manager and 2 developers of that sub-project.

The project uses a so-called architecture paradigm for the overall specification of the system. This is a new form of description in the organization, which has been developed by the method support department, but which has not been used before. The paradigm requires that a system is described from 5 perspectives, which together provide a detailed explanation of the product to be developed. The 5 perspectives are called product perspective, architecture perspective, development perspective, operation perspective and delivery perspective. As part of a feasibility study the project manager together with a system architect from the method support department and the product architects has produced such a description. It provides the scope for the 17 sub-projects through an overall requirement specification and an architectural description. Within these limits and those provided by the quality management guidelines the project managers of the sub-projects are free to choose their own development approach.

The architectural description for the project also defines the technical platform for the project. Project management together with the architects decided not to use the CASE tool and the accompanying method for component based development. They assessed that a standard financial and accounting system fitted the problem situation better. Therefore the system is developed towards such a system and large parts of the functionality are programmed with the help of a rule-based development tool. This has consequences for the utilization of methodologies in the project, as the method support department has no experiences and thus cannot offer assistance for this kind of development.
To communicate and coordinate the development across the different sub-projects the project uses UML based diagrams. In the sub-project, in which 3 of the interviews were carried out, the developers follow a method inspired by a form of object-oriented analysis and design chosen which one of the experienced developers had used before, but which had not been officially approved by project management for that project. The method was used with some adjustments to fit the particular needs of the project where the user requirements for the sub-project had already been defined and the design requirements were dictated by the decision to use the standard financial management system as a component.

As the overall architecture is new for the organization the project works closely together with the method support department, however in the particular sub-project investigated, no method consultant was used. Project C redevelops a specific part of another administrative system, which is used in the public sector. The system is based on a classical client-server architecture. As the system deals with an essentially independent functionality of a specific segment of a larger complex product group, it does not have to take into account existing systems and it is considered a pilot project to gather experiences with the development of business components for a particular business area. The project is close to being finished and it employs 20 people, of whom one project manager, who also is directly involved in the concrete development has been interviewed.

The project has tried to apply as closely as possible the CASE tool and the accompanying method. Thus, as a starting point the project developed the required business model, which from a user perspective describes work tasks, work-related concepts, events and user roles and the project used prototyping functions as provided by the tool to facilitate communication with the customer. The applied techniques - functional decomposition, data and process modeling as well as event and role analysis - and the resulting business model resemble very much techniques and results representing technology neutral and ‘true’ requirements as known from conventional structured analysis. Although it follows a method recommended by the method department, the project has not used a method consultant for its work.

**Interviewed Project Roles**

The 3 investigated projects all deal with the development of new systems. At the time of our research they had been on the way for between 1 and 2 years and the number of project participants varies between 12 and 20 project members. We interviewed people representing the following three roles in the projects: project managers, product architects, and system developers.

**Project Management** Project managers are in charge of the daily management of the project and have the final responsibility that the project lives up to the requirements posed to the system under development and that it stays within the given deadlines. All interviewed project managers are very experienced. They know the organization and the approaches, which are used to establish and to finalize projects very well. We interviewed 4 project managers.

**Product Architects** Product architects are in charge of the product’s architecture. The company’s large systems require a clear strategy for structure of a system and this is determined by the product architect. The product architects outline the product with respect to the project’s organizational, technical and market related context and have the responsibility for the system’s consistency with respect to the different requirements which derive from these. The product architects design the system’s component architecture; they map out which components the system should consist of and specify the overall interfaces between them. The product architects cooperate with the system architects, who are the method department representatives, to secure the general coherence between the organization’s differing systems. When the

*JITCA, Volume 6, Number 4*
architecture is determined the product architects follow up that the development actually complies with the decided architecture and support the developers in interpreting the architecture. In addition the product architects based are responsible for the overall documentation of the system. We interviewed 2 product architects.

System Developers System developers make the systems; they build the system’s different components and get them to work together. Developers analyze, design and implement the components on the basis of the determined architecture and the conditions defined by the project management. The developers are experts in the actual formation of the system and they determine how to work out the details of the product. The developers perform the actual programming and it is therefore their task to take care of the documentation which is directly related to the programmed components. In addition, the system developers test the components they have built. We interviewed 4 system developers.

WHAT DID WE FIND? – FIVE CATEGORIES OF FACTORS AND PROCESSES

On the basis of our empirical data we conclude that the practical use of system developments methodologies is influenced by factors and processes belonging to the following five categories: universality, confidence, experience, co-determination, and introduction. In the following we will describe each of these categories and present how we have found that the practical use of methodologies is affected in the projects.

Universality

In our study the idea of having one universally applicable methodology is not shared among the respondents. They argue that existing systems, which have been developed using a specific methodology and therefore their architecture and documentation were influenced by its methods, cannot be changed without paying respect to that methodology. As one project leader expresses it:

“... you cannot just use a totally new analysis method ... you at least have to take into account the method with which the program has originally been developed ...”

When developing large systems based on a sequential development model, which is promoted in the company for its generality, the developers have experienced problems with sequential methods. As the application domain is changing faster than it is possible to design and develop large and complicated systems, the initial analysis tends to be inaccurate, especially when the development runs over a longer period of time. One systems developer pointed out:

“Yes... Hmm... we gave that up! We found that we couldn't analyze 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!”

The complexity of the systems is also affecting the usability and the usage of the sequential methodology in the development projects. Some projects are so complex that the developers are not able to adequately comprehend all the dimensions at any one time. In some situations this leads to a paralytic state where the analysis just goes on and on without ever finishing. As one of the project leaders expressed 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.”
The Utilization of IS Development Methodologies in Practice

To avoid these situations some of the projects have adopted other methodologies than the official ones or supplemented them with alternative techniques. In project A none of the official methodologies is used at all and instead prototyping techniques are applied to make the process iterative and to develop the requested system.

Another reason why the one-size-fits-all-methodology is not feasible concerns the developers’ experience that different tools and platforms demand different methodologies. Software development in the company using the CASE tool always means applying the accompanying development guidelines. These however cannot be used without the tool. In project B the technical platform is a standard system combined with a business rule system and mainframe components. This directly affects the chosen methodology as, according to the project members, none of the formal methodologies fits this development setup. Therefore the project developed its own approach.

Finally, the application of the methods and techniques is influenced by the development tools. As one project leader formulated it: “Yes, the methods … When the company chose the CASE tool it also chose the methods, which came with it.”

In project C where development takes 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 modeling – instead of working with a more abstract business model as would be more sensible and as intended by the methodology underlying the guidelines. Beyond the CASE tools emphasis on technical design, a further problem is related to its separate dealing with analysis and design tasks. As a consequence the tool does not support automatic consistency checks when transferring the analysis model to the design model. This leads to extra work and is a further example for the development tool’s impact on the utilization of an ISD methodology.

Confidence

The developers explicated a need to feel confident about the progress of their work. Project C used a methodology allowing the development of early prototypes to help the developers sketch the business model of the system. These prototypes were not intended to represent any final system parts, and thus the customers are not presented with any user interfaces. The developers disliked this situation. They felt a need to please the customers and offered them a prototype where those could see the actual progress and got the possibility to recognize small parts of the final system. A project leader described the situation:

“... we had a reference group with some selected user ... these should be satisfied in one way or the other and see how the system would look like ...”

This would give the developers an indication or feedback that they were on the right track which created security for both sides. The project manager confirmed this by stating that seeing the customers applaud contributes to the drive of a project to sustain the development efforts.

The need for confidence was also brought up in the context of the larger, mission critical systems developed by the company. These are often under time pressure and the developers sometimes abridge the use of the prescribed methodologies. Typically the developers would like to collect and analyze more comments from the end users, but when the schedule does not allow this, the developers with a high degree of uncertainty develop solutions as they assume the end users want them. Not being able to use the techniques they
The Utilization of IS Development Methodologies in Practice

themselves have agreed upon, is experienced as unsafe and frustrating. Confidence is also an issue for management in the company. A project leader said:

"In the moment I'm writing a proper method document in the project group we known very well how we do things ... but management, likes to know that there is progress."

It also leads to a more symbolic use of the methodologies. In project A, which used the non-official prototyping approach, plans and communication upwards in the company were adjusted with the help of some of the official techniques to what the project leaders thought management needed with respect to feeling safe about the project's performance and continuity and lead to the desired effect. Providing management with some security is also the motivation for other projects, which use the CASE tool, but which do only so in a very superficial way. By applying prescribed standards these projects want to achieve goodwill from management.

Experience

The developers' experience with regard to the development process and the application domain influences method utilization. Experienced developers use their domain knowledge in many situations instead of the prescribed methods and analyze and design systems without going through formal step-by-step guides. One developer stated:

"... and then there are these nice diagrams. We skipped those. We knew there would be sequential events, so we just listed them in the right sequence."

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 toolbox where appropriate techniques and methods can be found. One architect explained:

"[...] 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 implications for the adoption of new methodologies as well. When experienced developers are not adopting the philosophy of a methodology their general way of working is not changing as a new methodology is introduced – only more tools and techniques are put in their toolbox. In contrast less experienced developers express a need for explicated methodologies, which help them learn the company's development practices. One young developer explained:

"... and then, how should I represent this. It is not that easy and you can't just say, we do it this or that way, how do we not this down, I think all this is much too much floating."

As this is not always the case, they have to rely on the experienced developers who thus have an important role and possess power with regard to the education of new developers. An experienced confirms this:

"... I am the one who has the most routine thus I influence the others in the project in a direction I have experience with ... this is because they are less experienced with it."
Co-determination

The use of methodologies in the development is also related to the developers’ desire for involvement and co-operation with management with respect to policy and decision making.

In project A with a high degree of such co-determination with regard to method selection — utilizing the prototyping approach – the motivation and eagerness for the project to succeed was much higher than in the other projects. The developers felt a responsibility, which resulted in a willingness to work very hard and it lead to an attitude where they wanted to show the rest of the organization that their way of doing things was good and worked well. With regard to involvement and motivation one project leader argued:

"Independently of how many estimation methods you have, there is always a little bit of guessing included. But if one has been involved in the guessing, one usually also wants to prove that the estimate holds. I think that that has much to say."

The developers’ urge to participate in the planning and implementation of their work conditions also lead to a methodological change as the project adopted the less formalized structure of extreme programming (XP), which appealed to the developers. The developers explain that they had heard of XP and that they wanted to try this approach. An architect stated:

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

Introduction

The way a methodology is introduced plays an important role with regard to its adoption. In the company with regard to the choice of development methods the formal requirements are less distinct and it is very much up to the projects to choose between the available methodologies. Having said this, a wish from management exists for the projects to adopt the component based development methodology. Of the three projects only one followed this methodology. One of the reasons for this lack of adoption is that management did not fully back up its introduction. One project leader argued:

“[…] 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 […]”

In addition, the developers were not offered the necessary education and training. The new tools were installed on the workstations and some courses were arranged, but there was no other information about why the new methodology was introduced. One project leader recalled:

“It was introduced in the following way ... We got to know that something new would come, and then we were send to some courses, and these confused us even more and that we just got to know that’s the way it now is.”

In contrast project B adopted a methodology to define the systems architecture. The project closely co-operated with the method support department and when the problems of understanding the techniques arose the method experts assisted and the adoption became less difficult. As a result a product architect experienced:

“I work much more structured now … now I have a shelf where I can put my things on …”
The Utilization of IS Development Methodologies in Practice

WHAT DID WE LEARN? – FOUR LESSONS LEARNED

The processes and factors defining the five categories which affect the practical use of system developments methodologies are not independent, but are highly interrelated. In the following we relate some of them to each other in more depth by formulating four lessons learned and linking these to the existing ISD literature.

Methodologies are adjusted in action; no universally applicable methodology exists

We have found the idea of having one universally applicable methodology flawed. Already Brooks (1987) describes that a methodology “silver-bullet” does not exist because of the varying complexity of information systems and Truex et al. (2000) state that ISD is locally situated. We also found that varying experience of the developers has an impact on the utilization of methodologies, which leads to the application of different methodologies in different projects.

The developers, too, express a close relation between the choice of development platform and the choice of methodology, which indicates that different methodologies are needed in different technical settings. They also experience limitations grounded in the implementation of methodologies in development tools. Like our respondents, Stolterman (1992) notices how developers become irritated when an approach does not credit them with a will and ability to judge and adjust methodologies to specific development situations and that they demand a certain element of freedom of choice. Nandhakumar and Avison (1999) report how a software tool forced the developers to improperly use specific techniques, and they confirm both the link between the choice of development tool and the choice of methodology and the experienced restrictions, which we found. All these findings render the claim of methodology universality void.

This is finally supported by the fact that developers’ curiosity and interest in new developments combined with their experience with inappropriate methodologies has an impact on the utilization of methodologies. In line with Fitzgerald (1998b) who states that working with new software technologies is a significant motivator among developers, our findings suggest that developers get inspired and like to apply new methodologies. Experimenting with new methodologies, is clearly in conflict with the idea of only using one all-purpose methodology, but is one of the developers’ driving forces for their work.

Methodologies are used symbolically

Our study provides examples of the use of methodologies with the purpose of making different parties more confident with regard to project progress. We confirm Wastell’s (1996) finding that, when developers are confronted with the challenge of developing large and complex information systems, a methodology can act as a means to provide them with comfort and confidence and to give them the belief that they are capable of the task.

Methodology utilization moves towards incremental methodologies

Our study supports that there is a move towards using methodologies proposing an incremental workflow and away from the sequentially organized methodologies. Size and complexity of the systems can become so large that developers are not able to cognitively comprehend them, and thus it is necessary to break them down in smaller parts or to start working with a smaller subset, e.g. in the form of prototyping. The same argument is valid for the customers or end users, they cannot comprehend the total system at once either (Curtis et. al., 1988). Like Fitzgerald (1998b) we find that the rapid changes in the application domain and business environment make it inappropriate to base development on the traditional life cycle approaches.

We extend that argument and provide empirical evidence of how an iterative approach gives the developers and the clients a sense of control and comfort as continuous deliverances or tests of prototypes illustrate.
The Utilization of IS Development Methodologies in Practice

progress to both parties. This point is further developed when also considering the extra motivation among developers when they are able to see that their work can be helpful to the clients.

Methodology adoption depends on management support, explicatons and co-operation

Like Kautz and McMaster (1994) we have found that poor management support makes it hard to diffuse new methodologies in an organization especially when no clear mission statement is communicated or when the necessary resources are not made available for the introduction process.

Our study suggests that it is problematic to educate staff in methodologies when these are not made explicit. In some of the investigated projects the methodology documentation was very sparse and thus, in the case organization, new employees had to rely on their more experienced colleagues. The latter are often more reluctant to adopt new methodologies, and thus the diffusion is slowed down.

We also demonstrate the importance of co-operation with the method support department. When the users and the designers of the methodology work closely together during method implementation, both parties benefit from the process. The developers understood the meaning of the provided techniques and the method designers improved their comprehension of the developers’ needs and thus redesigned parts of the method.

CONCLUSION

The objective of our research was to study people and their actions within an organizational context, namely whether and how system developers use ISD methodologies in practice. Based on an extensive empirical study we identified five interrelated themes of factors and processes, which affect the application of ISD methodologies, and formulated four lessons learned. The themes are described through the concepts of universality, confidence, experience, co-determination, and introduction. They emphasize that methodologies are adjusted in action and that no universally applicable methodology exists; that methodologies are used symbolically; that methodology utilization moves towards incremental methodologies; and that methodology adoption depends on management support, explications and cooperation. These results offer a framework for future research to get a deeper understanding of information systems development and the utilization of methodologies and to subsequently develop sustainable theories hereof. One way possible way of doing this is by considering the utilization of a methodology and the development process as innovation processes and applying an innovation research framework. By distinguishing between an individualist, a structuralist and interaction perspective on development practice this has already successfully been done in the field of software process improvement (Kautz and Nielsen, 2004). Finally, the framework can also be used by project managers and practitioners during actual development work to reflect upon and to constructively handle the utilization of a development methodology in the course of a project.

REFERENCES

The Utilization of IS Development Methodologies in Practice

22. Robey, D., and Markus, M.L., 1984, Rituals in information system design, *MIS Quarterly, 8*:*1*, pp. 5-15

*JITCA, Volume 6, Number 4*  19
The Utilization of IS Development Methodologies in Practice

Karlheinz Kautz, is a professor in Systems Development at Department of Informatics at the Copenhagen Business School, Denmark. His research interests are in system development practice and methodologies, knowledge management in the IT industry, software process improvement and the diffusion and adoption of information technology innovations. He has published in these areas in journals like Information and Software Technology, Information, Technology & People, the Scandinavian Journal of Information Systems, Software Process: Improvement and Practice, IEEE Software, Journal of Knowledge Management.

Bo Hansen, M. Sc, is PhD Student at the Department of Informatics, Copenhagen Business School, Denmark. His research interests are in knowledge management and software development practices, and he is currently involved in a national research project addressing in an action research manner software process improvement in the Danish IT industry. He has been employed as a systems engineer and developer in Danish Internet and investment companies.

Dan Jacobsen, M. Sc works as systems developer at Progressive IT, Denmark.