Reflections on Grounded Theory in Practice: Overcoming Obstacles in IS-related Studies

Paul Drews, Dorina Gumm
University of Hamburg, Department of Informatics
Vogt-Koelln-Str. 30, D-22527 Hamburg
{drews,gumm}@informatik.uni-hamburg.de
Working Paper, 31 August 2009

Abstract: In this paper we reflect on our experiences made in two IS-related research projects that were conducted with the Grounded Theory methodology. Although we esteem Grounded Theory, it shows some flaws that become discernible as recently as the researcher delves into analysis. Based on two research projects we discuss open coding, the consideration of context and process during theory building, as well as quality criteria and software usage to support the theory building process. Our experiences are summarised in seven lessons learned.

Introduction

Grounded Theory (GT) was originally developed by Glaser and Strauss as a research method in sociology (Glaser & Strauss, 1967). Like other methods it has been adopted successfully by IS researchers during the last decades. However, although the number of IS publications using GT has increased, it cannot yet be considered to be part of the IS mainstream: Only few publications are available that reflect on the use of the methodology for IS research (e.g. Urquhart, 2001; Hughes & Jones, 2003; Matavire & Brown, 2008). In addition, especially in the German tradition of Informatics, some professionals eye the use of sociological methodologies, and hence GT, suspiciously for informatics studies so that researchers have to justify the use of GT. This is our motivation to contribute to the ongoing discussion.

The authors have, independently from each other, chosen Grounded Theory as appropriate methodology for their IS-related research. The goal of this paper is to reflect the use of GT in our IS research projects and to identify clarifications, modifications and extensions turned out to be necessary in research practice. As a result, we present lessons we have learned from applying GT in our research. Accordingly, the presented insights base on in-depth reflection of research practice.

The paper is structured as follows: In the next section we give a brief overview on epistemological directions in IS and Informatics that allows a placement of GT in the methodological landscape of the respective research. After that we introduce our two research projects and indentify similarities and differences of the research designs. The main part of the paper is dedicated to experiences we made with GT. We reflect on methodological aspects that were most critical in our research processes. We conclude with a summary of the lessons learned and give suggestions for future improvements of GT use in IS research.

Grounded Theory in IS and German Informatics

Grounded Theory is a methodology for qualitative and interpretative research (Walsham 2006) which has its roots in social science and was developed by the sociologists Barney Glaser and Anselm Strauss in the 1960ies (Glaser & Strauss, 1967). The name Grounded Theory refers to theory that is derived from inductive analysis of and thus grounded in empirical data (Strauss & Corbin, 1990). It consists of a set of well-developed categories which are systematically interrelated to explain the phenomenon investigated (Strauss & Corbin, 1990; Coleman & O'Connor, 2007). A grounded theory is hence not "the formulation of some discovered aspect of a preexisting reality 'out there'"(Strübing, 2002, p. 321) but evolves during research and is interpretation depending on a given world view of the researcher (Strübing, 2002; Coleman & O'Connor, 2007).

For the reflection of GT in research practice, we refrain from a detailed (and necessarily superficial) description of GT, its coding procedures and analytical tools. Instead, in this section we discuss how GT as an empirical methodology fits into IS research and the German Informatics discipline.

However, discussing GT for IS studies is quite intricate for German researchers. IS is not an established discipline at universities in Germany. Instead, the main IT-related discipline is "Informatik" which can be compared to "informatics" as described by Bryant (2007, p. 127): Informatics "is the study of 'technologies of information and communication as well as the biological, social, linguistic and cultural changes that initiate, accompany and complicate their development'." This definition implies information systems as well as computer science or sub-fields like artificial intelligence and is hence similar to the German term "Informatik". Besides informatics, there are other IT-related disciplines, of which "Wirtschaftsinformatik" (business information systems) is also important for the background of the current paper. Both

profiles of informatics and business information systems overlap with IS research but cannot be considered comparable. GT is known as a research method in these disciplines but is rather seldom used (Wilde & Hess, 2007). We quoted the use of GT in IS studies in our research projects for legitimating GT in our contexts.

The choice of paradigm a researcher adopts depends mainly on the dominating "world view" in the discipline and the research problem; research design is thus determined by the fit between research problem and methodology (Coleman & O'Connor, 2007). As the definition of informatics by Bryant (2007) indicates, informatics has its roots in different scientific traditions. Research in this discipline follows different epistemological principles and hence applies different methodologies, respectively (GI 2009): For example, informatics as formal science roots in mathematics and focuses on formal descriptions of calculation methods; informatics as technical science focuses on general principles of development and handling of technology and has thus an engineering approach; informatics as design science focuses on the use context of software, the social aspects of software development, and thus on the interpretation of observation.

Empirical research is nowadays widely accepted in computer science and information systems (Basili, 1996; Mathiassen, 1998; Avison et al., 1999; Seaman, 1999; Galal, 2001; Dittrich et al., 2007) and varies according to the epistemological principles. Coleman & O'Connor (2007) identify two empirical research paradigms that receive most attention: the positivist paradigm which bases on the assumption of a reality outside our mental activity, and which is mainly related to quantitative methods and techniques; and the phenomenological or interpretative paradigm which assumes that truth is enacted (Strübing, 2002) and which is related to qualitative methods and techniques. Quantitative research follows a deductive research process, starting with hypotheses which can be tested, asking questions like how much?, how many?, how often?. In contrast, qualitative research follows an inductive research process in order to gather explanation and meaning, asking questions like how?, why?, in what way? (Coleman & O'Connor, 2007).
Although empirical research is widely accepted, and qualitative approaches are often adopted in informatics, purely qualitative studies are still scarce especially in technical-oriented research (Seaman, 1999; Sahay, 2003; Dittrich et al., 2007; Karlsson et al., 2007). However, empirical methods are also useful, for example in software development research, since software development is undeniably a social activity and thus implies aspects that purely

formal or technical perspectives are not able to cover (Seaman, 1999; Dittrich, 2002; Nørbjerg & Kraft, 2002). Qualitative methods allow gaining in-depth understanding of empirically studied social and cultural phenomena (Karlsson et al., 2007).

Meanwhile, Grounded Theory as qualitative and empirical approach to information systems research has gained acceptance (Avison et al., 1999; Beck, 2002) and has been successfully applied in a series of studies (e.g. Orlikowski, 1993; Galal & Paul, 1999; Sarker et al., 2001; Power & Moynihan, 2003; Rose et al., 2007; Coleman & O'Connor, 2007; Steen, 2007). They show that GT-based studies can provide valuable results for the IS discipline, and this is also true for more technical-oriented traditions of informatics or for business information systems studies.

Two Research Projects: Description and Comparison

Basis for our discussions are two research projects where GT has been applied as main research methodology. Although working in the same research group at our department, our fields of interests vary considerably, and our particular research projects did not overlap. As we have both chosen GT as appropriate methodology for our projects, we had the opportunity to reflect on benefits and obstacles of GT in two different research projects. In the following we briefly describe our projects and compare them according to research questions, epistemological concepts and methodological directions.

One research was conducted in the area of distributed requirements engineering (cf. e.g. Damian et al., 2003; Bhat et al., 2006; Berenbach, B., 2006). It aimed at understanding the interplay between distributed project settings and requirements engineering (RE) practice as an integrative part of the entire software development process. Goal of this research was to develop a model of distributed requirements engineering (DRE) that describes the characteristic challenges of distributed project settings and the interplay with requirements engineering practice in order to provide orientation for designing the development process. The result of the GT-based research process is a theory that identifies and conceptualises the main elements of DRE as well as immanent equivocalities, processes and interdependencies, and provides a new perspective on distribution.
Requirements engineering has "traditionally been positivist in its approach" (Hanisch & Corbitt, 2004, p. 3) when focusing on methods for elicitation, specification or validation of

requirements which are assumed to be the given reality that only needs to be documented. More attentive to qualitative and interpretative research is requirements *elicitation* in conjunction with evolutionary and participative development processes (Dittrich et al., 2007). However, such research typically focuses on “the context for the use of the (future) technology” (Beck, 2002, p. 162) and hence on the application of information technology, or on the understanding of software users’ practice while developing the software (Coleman & O’Connor, 2007). Social factors within the software development process itself are less considered and hardly in focus of qualitative and interpretative research (Coleman & O’Connor, 2007). In the following we refer to this project as "DRE project".

The other research project analyzed the innovation system for hospital IT in Germany. Its goal was to identify the socio-technical structure and the dynamics of innovation regarding the development and adoption of IT innovations in German hospitals. The theory developed in this research project provides a comprehensive overview of this sectoral innovation system, involved actors and IT. It elaborates on the innovation system for hospital IT, general structures of innovation projects and patterns of innovation in this context. Research on innovation has a long interdisciplinary tradition (Rogers 2003) and it is also established in IS research (e. g. Prescott & Conger, 1995; Fichman & Kemerer, 1999). While analyzing systems of innovation is often conducted with quantitative methods and on a national level (Nelson, 1993), research on sectoral innovation opens up a new perspective for IS researchers, as the interplay of development and use contexts plays a major role. Hospital IT is examined in IS research as well as in the medical informatics tradition. While the development and use of medical IT has unique characteristics, the exchange between IS and medical informatics is considered to be improvable (Chiasson & Davidson, 2002). In the following of this paper we refer to this project as "Hospital project".

Although these two research projects have rather different topics and research communities, they show certain parallels in their epistemological concepts. First, both research projects deal with socio-technical questions: One research reflects on requirements engineering as a technical task and a collection of engineering methods which influences and is influenced by a social (or: organisational, namely distributed) setting. The other research reflects on hospital IT in the context of social innovation processes.

Second, both research projects aim at understanding underlying processes and to develop theory that explains underlying structures and patterns of certain phenomena on a high level

of abstraction. With such a model they provide orientation knowledge as it is requested, for example, by Porto & Simon (2007) and Rolf (2008). Accordingly, both research projects did not started with hypotheses but with a more or less general research problem. Based on these epistemological interests, qualitative and interpretative research approaches have been selected for both research projects. Grounded Theory was chosen, as it provides a strong framework for developing theory on a high level of abstraction which is based on empirical data. It supports theory development by a "series of analytic tools and strategies" (Strauss & Corbin, 2008, p. 88) which guide the researcher through the process of data analysis.

Third, both projects followed the Straussian tradition of the GT (Corbin & Strauss, 1990). Whereas Glaser & Strauss's (1967) version of GT emphasises that theory "emerges" from data without a too formalistic analysis process, Strauss & Corbin (1990) created an extended coding system which supports the researcher's theory building. Another fundamental difference is that in Glaser & Strauss's (1967) version the research problem and question can only be discovered during the coding process. In contrast, Strauss & Corbin believe that "a question should be preset as it sets the boundaries around the study area" (Coleman & O'Connor, 2007, p. 656). Accordingly, their version takes into account the researcher's previous knowledge of the phenomenon to be studied, as well as that literature and professional experience cannot be completely avoided before a study starts -rather, this supports theory building and theoretical sensitivity. For both our studies the Strauss & Corbin's (1990) approach is more appropriate for the research conditions: In order to collect relevant empirical data, it was necessary to have a somewhat specific research problem. Stating a research problem, in turn, requires an extensive knowledge of existing research in the related area. The data source used for analysis in both studies consists of a series of semi-structured expert interviews (Charmaz, 2003) which have been completely transcribed for the coding processes.

Fourth, in both projects literature was used as a starting point for the interview guidelines. Grounded theory is used to build up a new theory out of the data. But before starting data gathering, he or she should have an understanding of the state-of-the-art in the respective research field. GT speaks about theoretical sensitizing and supposes to use existing literature in the theory building process (Corbin & Strauss, 2008). We intensively used this idea of sensitizing and literature played an important role for the development of our interview guidelines.

In the Hospital project, for example, literature describes innovation and knowledge barriers as factors that hinder innovation processes (Attewell, 1992; Fichman & Kemerer, 1997; Rogers, 2003; Hauschildt, 2004) and lead users to play an important role (Hippel, 1988; Hippel, 2005). Consequently, the interview guidelines included questions to find out, which barriers exist in innovation processes for hospital IT and who fosters innovation most intensively. On the one hand, existing theory may quickly guide the interviewer and the interviewee to relevant and important aspects. On the other hand, the coding process afterwards should not directly adapt the concepts of preexisting theory.

Differences in the GT application exist regarding the iterative and incremental process of data collection and data analysis as well as in the use of software and additional data sources. The DRE project started with a pre-study with a series of very small interviews, combined with a little questionnaire and a collection of observations. This pre-study was used to sharpen the research focus. As one result of the pre-study, the author developed a taxonomy for the sub-topic of distribution on the basis of a literature study. After that the actual GT-based research process started. The first two interviews were conducted at intervals of three months. However, because of some coding difficulties (cf. next section), most of the interviews in this project were conducted and transcribed half a year later consecutively during a period of seven weeks before they were coded in detail. In addition to the interview transcriptions, the data collected during the pre-study as well as single documents provided by the interviewees were included in the coding processes.

In contrast, the hospital IT-project directly began with the data collection. The interviews were spread over a period of two years and followed an incremental approach. Furthermore, additional data sources like non-scientific journal articles, observation protocols, and websites were used as data sources. Software for computer aided qualitative data analysis (CAQDAS) supported the data analysis (Lewins & Silver, 2007).

Reflecting the use of GT

Grounded Theory is a very work-intensive methodology, as Strauss and Corbin and other researchers emphasize. We can confirm this. This is true especially for the axial and selective coding process where it is necessary not only to think about codes and their relationships, but, again and again, constantly compare developed categories and raw material. Despite that, or actually because of it, Grounded Theory is a very powerful methodology. We have come to

appreciate GT in quite different research projects, because it supports data analysis and theory building with valuable coding procedures and other analytic tools for guiding through the analysis process. The methodology is a very good instrument when developing theory that is very well grounded in empirical data.

However, we have also faced difficulties and made experiences that demand reflection. Firstly, open coding, although the "simplest" of the coding procedures, can hold some traps that are related to existing scientific work. Secondly, the consideration of context is not well supported by Grounded Theory. Thirdly, also for the consideration of process, we missed a structured guidance for analysis and conceptualisation. Fourthly, quality criteria need special attention in IS studies. Fifthly, software usage turned out to increase productivity and facilitate the "grounding" of the theory. In the following paragraphs we reflect on these experiences in detail.

Open coding

Open coding is the first coding procedure in the analysis process. In this step basic and "simple" concepts in the data are identified which label particular observations. A researcher should be - although having a sound scientific background - open-minded during analysis and conceptualisation. This is not simple, as so-called a priori codes (Lewins & Silver, 2007) are easily taken: In the Hospital project, people and organizations were coded as "actors" in the beginning. By referencing this elaborated concept of sociology (Rammert, 1993), the researcher immediately has to deal with the question whether the theoretical underpinnings of this concept match with the data, and, if different conceptualizations of this concept in theory exist, which one might fit. In this project, the researcher decided to step back from the "actor" concept and chose codes, which are closer to the data (in-vivo codes) like "hospital", "IT department", or "developer". This example demonstrates a challenge in GT-based research processes: How far should existing theory be used during the coding phases? On the one hand, literature and experience should be used to increase the "theoretical sensitivity" (Strauss & Corbin, 2008, p. 230). On the other hand, a direct link to existing theory immediately brings up the question regarding the relationship of data, existing theory and the theory to be developed.

This problem was even larger in the DRE study. The interviews focused on requirements engineering practice - an area that is well described and conceptualised in literature. The

study's author and the interviewees were more or less familiar with the established terms. Even when using in-vivo codes (Corbin & Strauss, 2008), the codes remained too close to well known conceptualisations, although the author reflected on this and questioned such codes. The recommendation by Strauss & Corbin to use one-word codes for labelling phenomena, especially to avoid descriptive summaries of observations, turned out to intensify this problem. Codes emerging in the open coding process were, for example, task, tool, customer, requirement, elicitation, distance, communication. Similar experience was made with the second part of the research topic, distributed software development. It was not possible to abstract from these codes and to get insights beyond existing knowledge about requirements engineering or distributed software development. The problem was that one-word codes were not appropriate to get close to the actual research focus, the interdependencies between both topics. This problem was addressed in two ways: First, instead of using one-word codes, the researcher searched for phenomena that were received as problematic by the interviewees, hence for unusual or unexpected circumstances. For this step it was necessary to code large parts of interviews which resulted in codes with several words (almost sentences) describing entire situations. These were later grouped (categorised) using one- or two-word codes. However, starting with this kind of coding, the categorisation and conceptualisation of these codes turned out to be difficult with only two interviews. Second, instead of having an alternating process of data collection and analysis, further interviews in quick succession had been conducted. Goal of this approach was to have a collection of open-coded phenomena from different interviews which allowed for finding underlying concepts in the identified unusual or unexpected circumstances which could not be found by consulting only one or two interviews.

Open coding is a valuable and necessary procedure, and a priori codes can be helpful, However, our conclusion here is to avoid direct use of theoretical concepts and discussions the use of such concepts would entail. The codes should be close to the data, and if an existing concept is referenced, the researcher should be aware that her or his data might be incompatible with this concept (Corbin & Strauss, 2008, p. 38). A comparison between the newly built grounded theory and existing theory is of course an important aspect, but it should be done in a later research phase. In addition, when working on a research project in an area of well-established terms, where interviewees use al lot of words established in related literature, it can be helpful to diverge from the given methodology.

Considering context and situation

Though central elements of the GT toolset turned out to be helpful during the process of theory generation, this is not the case for the conditional matrix (Strauss & Corbin, 1990; Corbin & Strauss, 2008). It is meant to support the process of finding answers to the questions "who, what, where, when, and why" (Corbin & Strauss, 2008) and to analyse different levels between a single action and the global level (Strauss & Corbin, 2008). In both of our studies the conditional matrix did not provide the support needed for drawing connections between the social process and its relevant contexts.

First, IT artefacts are an essential aspect of IS research, and their diverse nature (Orlikowski & Iacono, 2001) and role in social processes (Orlikowski, 1992; Orlikowski, 2000) need to be considered in theory building. GT does not provide the appropriate support for this yet. Second, in both studies, the context of the explored social processes (innovation and requirements engineering practice) played an important role. In the Hospital project the context was focused on IT for German hospitals. Therefore it had to take a large part in the theory: The first structural part of the Hospital project's theory describes the "system of innovation" under consideration. It introduces actor, arenas, IT-systems, relevant standards, labour and its division among involved actors, spatial structures, social discourses, and a historical perspective. These structural elements are later on used to connect general aspects of innovation projects (e.g. innovation barriers) with the context (e.g. certain actors or IT-systems). Without this link between the system of innovations' structure and the social process the theory would lack precision in the relationship between action and its context.

In the DRE project the context focused on distributed project settings. Because the phenomenon of distribution in software development was extensively discussed in literature but not well conceptualised when conducting the study, a taxonomy for this phenomenon was developed. This taxonomy was used as discussion framework in the interviews and later related to the concepts and categories developed by using the GT coding procedures.

Like Clarke (2005) we come to the conclusion that the conditional matrix is one of the largest weaknesses of GT. Though Clarke's Situational Analysis tries to resolve this problem by providing additional analytical concepts, a new tool should be developed that gives more detailed instructions to the researcher than Strauss and Corbin's matrix. But for IS research,

Clarke's concepts should be refined and supplemented with IS specific contextual parameters, especially regarding the IT artefact and its social embeddedness.

Considering Process

Closely related to integrating the context is linking concepts and categories to processes. Grounded Theory provides the idea of process and the conditional matrix to identify processes in the empirical data. Process is a powerful perspective for analysis: the conceptualisation of process-like incidents explains why routines collapse, why problems occur, or why one detects growth, development, movement or failures and stagnation when looking back (Strauss & Corbin, 1990). According to Strauss & Corbin (1990) and Strauss et al. (1985), processes can be described in two different ways: As a sequence of phases or stages, or as a non-progressive movement. Whereas a sequence of phases describes a process from a starting to an end point in time, a non-progressive movement is a description of general movements or processes, called trajectory. In their book about the social organisation of medical work, Strauss et al. (1985, p. 5) exemplify this distinction: The course of illness is an example for the first kind of process; it describes how an illness gradually develops "over time, getting worse and then perhaps clearing up". The term trajectory refers "not only to the physiological unfolding of a patient's disease but to the total organization of work done over that course plus the impact on those involved with that work and its organization".

However, the identification and conceptualisation of processes were very difficult, because GT does not provide a toolset for identifying and conceptualising processes as it does for concepts and categories. Both books by Corbin & Strauss (1990, 2008) only give some study examples but no methodological guidance. This was for both research projects unsatisfying.

In the DRE project two other concepts had been consulted: The technology use path is a concept used in the area of technological impact assessment (Funken, 2000; Rolf, 2008) and path theories (e.g. Schreyögg & Sydow, 2003; Ebbinghaus, 2005). It is a tool to analyze the history of technology use and respective decisions. The technology use path traces technology innovations on the individual, organisational or societal level by following the path of usage. The "path" metaphor denotes that the process is not determined but also implies (former) possible branches, alternatives and options. The technology use path analysis allows to understand why things happened although convincing alternatives were maybe available (Rolf, 2008). The technology use path exemplifies what Corbin and Strauss term a sequence of phases or stages.

For identifying trajectories, Weicks explanations on processes of organising and the introduction of causal loops turned out be helpful. Causal loops – basing on research in organisational studies – as introduced by Weick (1995), conceptualise interdependencies between variables, a concept that allows abstracting from actors, organisations, activities or any other type of entity. Causal loops are given if variables affect each other. One example Weick gives is the following causal loop: An increasing number of inhabitants of a city increases the amount of waste per unit of area; this in turn increases the bacteria per unit of area; this increases the amount of diseases and this decreases the number of inhabitants of the city. The presented loop is called deviation-neutralising since the increase of the amount of inhabitants provokes eventually the decrease of the amount of inhabitants. With causal loops ambiguities are unravelled if entities are not understood as causes and effects of each other but as interdependent variables. It is then an arbitrary decision to call one entity cause and another effect. Causal loops also allow illustrating underlying processes. They have not a defined beginning or end; rather, the respective effects influence each other to infinity.

As GT lacks in guidance for analysing and conceptualising processes, the ideas of technology use path and causal loops helped a lot for the research. However, both concepts, admittedly, do not provide a toolset for analysis like the coding procedures of the GT do. It would be very helpful, especially for researchers that are not domiciled in sociology, if GT would provide a toolset for process analysis.

Quality criteria

Evaluating the quality of qualitative research is not easy (Strauss & Corbin, 2008, p. 297). While the terms "validity and reliability" were once used be Strauss & Corbin (1990), Juliet Corbin now avoids these terms as they carry "too many quantitative implications" (Corbin & Strauss, 2008, p. 301) and suggests "credibility" as a more appropriate term. Deciding whether a GT is considered to be plausible is subject to the decision of participants, researchers and readers (ibid, p. 302). Therefore, researchers should present and discuss their results not only in the research community but also with the participants of the study. In our projects this was achieved by integrating ad-hoc-hypotheses in the interviews (Hospital project) and by discussing preliminary results with some of the interviewees (both studies).

Strauss and Corbin name further conditions that foster quality (ibid, p. 302) of which "methodological consistency" deserves a close look. It refers to the demand of applying a

method completely and that it should not be combined with other methods ("method slurring"). While our studies kept close to the GT tools and strategies (coding procedures, constant comparison, paradigm, etc.) we also faced some difficulties (see above) that necessitated modifying them. Is this consistency or slurring? On the one hand, we departed from the pure GT when considering situational analysis, watered down the alternation of data collection and analysis, or consulted additional theories for process identification. On the other hand, this was only done because of shortcomings in the method. The constant reflection of the method's use is reasonable as it is not an established method in IS research yet.

Theories can also be differentiated regarding their generality. In quantitative research this is connected with statistical significance, which is not appropriate for qualitative research (Steen, 2007). Strauss & Corbin (1990) emphasise the explanatory power of grounded theories rather than their generalisability. Grounded theories are context-dependent, and thus they are at best generalisable within the context and boundaries of a particular study. Strauss & Corbin (ibid) also state that generality of a theory increases, the more systematic the theoretical sampling is, and the more conditions and variances are included in the theory. Weick (1995) elaborates (referring to Thorngate's (1976) postulate of commensurate complexity) that generality, accuracy and simplicity are each virtues of research results, but altogether they are disparate to achieve. Weick illustrates this with a clock face (see Figure 1). Research that aims for generality and accuracy (2 o'clock) will be rather complex and forego Occam's razor. Research that aims for accuracy and simplicity (6 o'clock) will be less generalisable and focuses on situational interpretations. Research that aims for a general and simple description of a phenomenon (10 o'clock) will lack some accuracy, they are easy to understand but difficult to apply accurately for a specific case. Weick underscores that each type of research is equally valuable. He also suggests that researchers should keep in touch with research with the opposite focus. That means that 6-o'clock-research needs to be related to research which is more general.

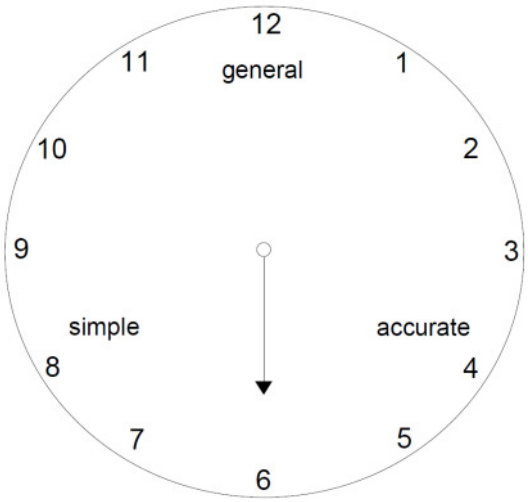

Figure 1 Classifying research with Weick's clock face

The research from our studies can both be called 6-o'clock-research. Firstly, they are based on empirical data and consider the context and is thus situational research. Secondly, the theory building needs a special focus in this research. This is achieved by using the selective coding procedures of the GT. The opposite research is included in both studies by relating the research questions and the results to the Mikropolis model (Rolf, 2008). While the Mikropolis model provides a framework for studying general and global phenomena of IT development and use, the two grounded theories provide detailed and contextualised concepts. But the two research projects also show differences regarding the generalisability of their results. The DRE project presents categories that are rather general phenomena like "organisational interfaces" The findings could be observed independently from the business sector, type of project or role of the interviewee within the project. Therefore, the results of this research are expected to be valid in other projects that deal with distributed requirements engineering activities. The Hospital Project, however, has a much stronger link to a certain context. The three parts of grounded theory build in this project – innovation system, innovation project, and patterns of innovation – vary in their degree of generalisability. While the description of the innovation system is "pure" context and the project part is expected to be valid in other sectors, the innovation patterns can be located in between. Though both research results are contextualised, the degree of expected generalisability varies. It depends on the relationship of research question and empirical data as well as on the degree of generality that is aspired.

Software usage

In the DRE project no software for data analysis was used. This lack of software support made the process of theory building very difficult. To deal with codes, concepts and categories is quite easy with text processors. With stern discipline, the same is true for

keeping traceability between raw data, emerging codes and concepts. But what is really difficult is to gain an overview over all the interdependencies and relationships between codes, concepts, categories. It was an incredible amount of work, time and mental activity to develop these relationships and not to loose track in all the data. The usage of appropriate software would have made the process of theory building much easier.
The Hospital Project started without using software support. But after having begun with the open coding it turned out that keeping track of the relationships between primary documents, codes and memos is very time consuming. Therefore an evaluation of computer aided qualitative data analysis software on the basis of Lewins and Silvers' (2007) guide was conducted. All three tools (NVivo, Atlas.ti and MaxQDA) mentioned in the guide were tested. Finally, the choice was made for Atlas.ti as it provides the greatest flexibility for building relationships among the different items. As all three lacked appropriate support for PDF documents, several empirical documents needed to be summarised in new documents so that they could be directly coded. Until now the software packages improved support for PDF documents. Because the software allows a very quick access to the primary documents via codes or later on via memos, the productivity of data management increased. Also, the housekeeping among the large number of codes (e.g. merging of codes) that is generated in the open coding phase is much easier with the software support. During coding, memo writing and the final report writing, the quick and easy access to codes and data supported the process of "grounding" the theory in the data (Strauss & Corbin, 1990; Corbin & Strauss, 2008). As Lewins and Silver (2007) point out, the use of software does not replace the intensive internalisation and use of qualitative methods. But it can increase researchers' productivity, support the coding processes and foster the "grounding" of theory.

Conclusion

In this paper we have presented two IS research projects for which Grounded Theory had been applied. On the basis of these two projects we reflected on the methodology's application by focussing on aspects that turned out to be most critical for our research. These aspects are: open coding, the consideration of context and process during theory building, and quality. In the following paragraphs we summarise these experiences as seven lessons learned

1. Interrupt analysis if you start to match theoretical underpinnings of a code with empirical data. For example, if the code "actor" is used and questions arise whether theoretical concepts of this term fit to the data and vice versa, this is an indication that the

researcher "lost her or his open mind". At this point, other, maybe less abstract or in-vivo codes should be used.

2. Dare to modify coding advices if necessary. For example, if interviewees are familiar with theoretical concepts of the topic discussed, one-word and especially in-vivo codes can be a hindrance to analysis. In this case, a more descriptive kind of coding can be helpful.

3. Combine qualitative analysis of empirical data with other kinds of model building for theory development. In research projects where the actual research and empirical study needs to be placed in a specific context, this context needs a careful examination. This can go beyond the empirical data and thus has to be done on the basis of literature. In order to develop a coherent theory, the grounded theory and the context must be connected. The conditional matrix, the main GT tool for context consideration, is not satisfying.

4. For systematically developing processes in the theory, consult other theories or methodologies. Grounded Theory does not provide systematic guidance for identifying, analysing and conceptualising process. Path theories and causal loops turned out to be helpful, especially as they fit to the particular research question and context of the DRE project. However, as long as Grounded Theory does not provide better guidance for process analysis, a researcher should consult other resources that are appropriate for their projects.

5. The quality of a resulting grounded theory should be evaluated by the consistency and explanatory power. It is not sufficient to evaluate it (exclusively) by the pureness of the methodological research approach. As our reflections show, consulting other methodological guidance as well as a slightly modifying GT tools were necessary in both projects.

6. Reliability and validity remain important criteria in (business) informatics. Validity and reliability as quality criteria are difficult to apply to any qualitative research result and especially to GT studies. More appropriate is to evaluate the results in respect of internal consistency and explanatory power. However, reliability and validity or generality remain important criteria to evaluate and defend research results in (business) informatics – at least in the German research tradition. Despite of that, if evaluating a grounded theory against such criteria, these have to be discussed in the context of qualitative research.

7. Usage of CAQDAS recommended. Such software increases the productivity of the analysis process. It supports the traceability of raw data and emerging concepts, the housekeeping among a large number of codes and the easy access to data and codes.

However, apart from the discussed difficulties, we experienced Grounded Theory as a very valuable methodology for our research projects. As both projects dealt with socio-technical phenomena, aimed at developing a fundamental model of particular phenomena and did not start with hypothesises to verify, Grounded Theory fitted very well. In both research projects, theories could be created that consist of well developed categories which could be systematically related to each other, while categories and their relations are solidly grounded in empirical data. We can in good conscience recommend Grounded Theory for IS studies but would advise not to use GT in a too dogmatic way. It is always important to reflect the tools of a methodology against actual research requirements and, if necessary, to modify or enhance the methodological toolset.

References

AVISON DE, LAU F, MYERS MD and NIELSEN PA (1999) 'Action research'. *Communication of the ACM* 42(1), 94-97

ATTEWELL P (1992) Diffusion and Organizational Learning: The Case of Business Computing. *Organizational Science* 3(1), 1-19

BASILI VR (1996) The role of experimentation in software engineering: Past, current, and future. In *Proceedings of ICSE-18*, pp. 442–449

BECK EE (2002) What doesn't fit: the "residual category" as analytic resource. In *Doing empirical research on software development: finding a path between understanding, intervention, and method development*, (DITTRICH Y et al., Eds), MIT Press, Cambridge, MA, USA

BERENBACH B (2006) Impact of organizational structure on distributed requirements engineering processes: lessons learned. In *GSD '06: Proceedings of the 2006 international workshop on Global software development for the practitioner*, ACM Press, New York, NY, USA, pp. 15–19.

BHAT JM, GUPTA M and MURTHY SN (2006) Overcoming requirements engineering challenges: Lessons from offshore outsourcing. *IEEE Software*, Special Issue on GSD 23(5), 38–44

BRYANT A (2007) *Thinking "Informatically": A New Understanding of Information, Communication, and Technology*. Edwin Mellen, Lewiston.

CHARMAZ K (2003) Qualitative Interviewing and Grounded Theory Analysis. In (HOLSTEIN JA, GUBRIUM JF, Eds) Inside Interviewing – New Lenses, New Concerns. Sage Publications, Thousand Oaks, CA

CHIASSON M and DAVIDSON E (2002) Getting the Two to Dance: Examining Barriers To Health Information Systems Research in Mainstream IS journals. In *Proceedings of th 35th Hawaii International Conference on System Sciences*, Hawaii, IEEE

CLARKE AE (2005) *Situational Analysis - Grounded Theory after the postmodern turn.* Sage, Thousand Oaks

COLEMAN GO and O'CONNOR R (2007) Using grounded theory to understand software process improvement: A study of Irish software product companies. *Information and Software Technology* 49(6), 654–667

CORBIN J and STRAUSS A (2008) *Basics of Qualitative Research*, 3rd Edition. Sage Publications, Thousand Oaks.

DAMIAN DE, EBERLEIN A, SHAW ML and GAINES BR (2003) An exploratory study of facilitation in distributed requirements engineering. *Requirements Engineering* 8(1), 23–41

DITTRICH Y, JOHN M, SINGER J and TESSEM B (2007) Editorial: For the special issue on qualitative software engineering research. *Information and Software Technology* 49(6), 531–539

EBBINGHAUS B (2005) Can path dependence explain institutional change? Two approaches applied to welfare state reform. *MPIfG Discussion Paper* 05/2, Max Planck Institute for the Study of Societies Cologne.

FICHMAN RG and KEMERER CF (1997) The Assimilation of Software Process Innovations: An Organizational Learning Perspective. *Management Science* 43(10), 1345-1363

FUNKEN C (2000) *Modellierung der Welt - Wissensoziologische Studien zur Software-Entwicklung*. Leske + Budrich, Opladen.

GALAL GH (2001) From contexts to constructs: the use of grounded theory in operationalising contingent process models. European Journal of Information Systems 10 (1), 2-14

GALAL G and PAUL R (1999) A qualitative scenario approach to managing evolving requirements. *Requirements Engineering* 4(2), 92–102

GI (2009) Was ist Informatik? Unser Positionspapier. http://www.gi-ev.de/fileadmin/redaktion/Download/was-ist-informatik-lang.pdf, last visited 30-AUG-2009.

GLASER BG and STRAUSS A (1967) *The Discovery of Grounded Theory: Strategies for Qualitative Research*. Aldine Transaction, Chicago.

HANISCH J and CORBITT B (2004) Requirements engineering during global software development: Some impediments to the requirements engineering process – a case stud. In Proceedings of the European Conference on Information Systems ECIS'04

HAUSCHILDT J (2004) *Innovationsmanagement*. Vahlen, München.

HIPPEL EV (1988) *The sources of innovation*. Oxford University Press, New York ; Oxford.

HIPPEL EV (2005) *Democratizing innovation*. MIT Press, Cambridge, Mass.

HUGHES J and JONES S (2003) Reflections on the use of Grounded Theory in Interpretative Information Systems Research. In *Proceedings of the 11th European Conference on Information Systems, ECIS 2003*, Naples, Italy 16-21 June 2003

KARLSSON L, DAHLSTEDT ÅG, REGNELL B, NATT OCH DAG J and PERSSON A (2007) Requirements engineering challenges in market-driven software development – an interview study with practitioners. *Information and Software Technology* 49(6), 588–604

LEWINS A and SILVER C (2007) *Using Software in Qualitative Research – A step-by-step Guide*. SAGE Publications, London.

MATAVIRE R and BROWN I (2008) Investigating the Use of „Grounded Theory" in Information Systems Research. In *Proceedings of SAICSIT 2008*, 6.-8. October 2008, Wilderness, South Africa.

MATHIASSEN L (1998) Reflective systems development. Scandinavian Journal of Information Systems 10(1-2), 67–117

NELSON RR (1993) National Innovation Systems: A Comparative Analysis. Oxford University Press.

NØRBJERG J and KRAFT P (2002) Software practice is social practice. In *Doing empirical research on software development: finding a path between understanding, intervention, and method development*, (DITTRICH Y et al., Eds), MIT Press, Cambridge, MA, USA, pp. 205–222

ORLIKOWSKI WJ (1992) The Duality of Technology: Rethinking the concept of Technology in Organizations. *Organization Science* 3(3), 398-427

ORLIKOWSKI WJ (1993) CASE tools as organizational change: Investigating incremental and radical changes in systems development. *MIS Quarterly* 17(3), 309–340

ORLIKOWSKI WJ (2000) Using Technology and Constituting Structures: A Practice Lens for Studying Technology in Organizations. *Organization Science* 11(4), 404-428

ORLIKOWSKI WJ and IACONO CS (2001) Research Commentary: Desperatly Seeking the "IT" in IT Research - A Call to Theorizing the IT Artifact. *Information Systems Research* 12(2), 121-134

PRESCOTT MB (1995) Diffusion of Innovation Theory: Borrowings, Extensions, and Modifications from IT Researchers. *Data Base Advances* 26(2&3), 16-19

PRESCOTT MB and CONGER SA (1995) Information Technology Innovations: A Classification by IT Locus of Impact and Research Approach. *Data Base Advances*, 26(2&3), 20-41

POWER N and MOYNNIHAN T (2003) A theory of requirements documentation situated in practice. In *SIGDOC '03: Proceedings of the 21st annual international conference on Documentation*, ACM, New York, NY, USA, pp. 86–92

PORTO DE ALBUQUERQUE J and SIMON EJ (2007) Dealing with socio-technical Complexity: Towards a transdisciplinary Approach to IS Research. In (ÖSTERLE H, SCHELP J, WINTER R, Eds) Proceedings of the 15th European Conference on Information Systems (ECIS 2007), 2007, S. 1458-1468.

RAMMERT W (1993) *Technik aus soziologischer Perspektive Forschungsstand, Theorieansätze, Fallbeispiele ein Überblick*. Westdeutscher Verlag, Opladen.

ROGERS EM (2003) *Diffusion of Innovations*. Free Press, New York.

ROLF A (2008) *Mikropolis 2010. Menschen, Computer, Internet in der globalen Gesellschaft*. Metropolis Verlag, Marburg.

ROSE J, PEDERSEN K, HOSBOND JH, and KRÆMMERGAARD P (2007) Management competences, not tools and techniques: A grounded examination of software project management at WM-data. *Information and Software Technology* 49(6), 605–624

SARKER S, LAU F and SAHAY S (2001) Using an adapted grounded theory approach for inductive theory building about virtual team development', *SIGMIS Database* 32(1), 38–56

SAHAY S (2003) Global software alliances: the challenge of standardization. *Scandinavian Journal of Information Systems* 15(1), 3–21

SCHREYÖGG G and SYDOW J, Eds (2003) *Strategische Prozesse und Pfade, Managementforschung 13*. Wiesbaden, Gabler.

SEAMAN CB (1999) Qualitative methods in empirical studies of software engineering, *IEEE Transactions on Software Engineering* 25(4), 557–572

STEEN O (2007) Practical knowledge and its importance for software product quality. *Information and Software Technology* 49(6), 625–636

STRAUSS AL and Corbin J (1990) *Basics of Qualitative Research: Grounded Theory Procedures and Techniques*. Sage Publications, Sage Publications, Newbury Park, CA.

STRAUSS A, FAGERHAUGH S, SUCZEK B and WIENER C (1985) *Social Organization of medical Work*. Transaction Publishers, New Brunswick (U.S.A).

STRÜBING J (2002) Zum Konzept der Herstellung und Sicherung von Qualität in grounded theory-basierten Forschungsarbeiten. *Kölner Zeitschrift für Soziologie und Sozialpsychologie* 54(2), 318–342

THORNGATE W (1976) "In general" vs. "it depends": Some comments on the Gergen-Schlenker debate. *Personality and Social Psychology Bulletin* 2, pp. 404-410URQUHART C (2001) An Encouter with Grounded Theory: Tackling the Practical and Philosophical Issues. In (TRAUTH EM, Ed) Qualitative Research in IS: Issues and Trends, Idea Group Publishing, Hershey, pp. 104-140

WALSHAM G (2006) Doing interpretive research. *European Journal of Information Systems* 15(3), 320-330

WEICK KE (1995) *Der Prozess der Organisierens* (english title: The Social Psychology of Organizing), suhrkamp taschenbuch wissenschaft.

WILDE T and HESS T (2007) Forschungsmethoden der Wirtschaftsinformatik - Eine empirische Untersuchung. *Wirtschaftsinformatik* 49 (4), S. 280-287

ZÜLLIGHOVEN H (1998) *Das Objektorientierte Konstruktionshandbuch*. Dpunkt Verlag.